University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

Dissertations, Theses, and Student Research Papers in Mathematics

Mathematics, Department of

Fall 12-1-2015

Analysis of Neuronal Sequences Using Pairwise Biases

Zachary Roth
University of Nebraska-Lincoln, s-zroth1@math.unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/mathstudent

 Part of the Applied Mathematics Commons, and the Computational Neuroscience Commons

Roth, Zachary, "Analysis of Neuronal Sequences Using Pairwise Biases" (2015). *Dissertations, Theses, and Student Research Papers in Mathematics*. 68.
http://digitalcommons.unl.edu/mathstudent/68

This Article is brought to you for free and open access by the Mathematics, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Dissertations, Theses, and Student Research Papers in Mathematics by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

ANALYSIS OF NEURONAL SEQUENCES USING PAIRWISE BIASES

by

Zachary J. Roth

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Doctor of Philosophy

Major: Mathematics

Under the Supervision of Professor Vladimir Itskov

Lincoln, Nebraska

December, 2015

ANALYSIS OF NEURONAL SEQUENCES USING PAIRWISE BIASES

Zachary J. Roth, Ph.D.

University of Nebraska, 2015

Advisor: Vladimir Itskov

Sequences of neuronal activation have long been implicated in a variety of brain functions. In particular, these sequences have been tied to memory formation and spatial navigation in the hippocampus, a region of mammalian brains. Traditionally, neuronal sequences have been interpreted as noisy manifestations of neuronal templates (i.e., orderings), ignoring much richer structure contained in the sequences. This paper introduces a new tool for understanding neuronal sequences: the bias matrix. The bias matrix captures the probabilistic tendency of each neuron to fire before or after each other neuron. Despite considering only pairs of neurons, the bias matrix captures the best total ordering of neurons for a sequence (Proposition 3.3) and, thus, generalizes the concept of a neuronal template.

We establish basic mathematical properties of bias matrices, in particular describing the fundamental polytope in which all biases reside (Theorem 3.25). We show how the underlying simple digraph of a bias matrix, which we term the bias network, serves as a combinatorial generalization of neuronal templates. Surprisingly, every simple digraph is realizable as the bias network of some sequence (Theorem 3.34). The bias-matrix representation leads us to a natural method for sequence correlation, which then leads to a probabilistic framework for determining the similarity of one set of sequences to another. Using data from rat hippocampus, we describe events of interest and also sequence-detection techniques. Finally, the bias matrix and the similarity measure are applied to this real-world data using code developed by us.

COPYRIGHT

© 2015, Zachary J. Roth

DEDICATION

To my family, thank you. My mother and father both worked hard and without complaint in raising me despite not receiving anything in return, and they each strived to live well as examples for me. Mom, much of my personality and humor stems from you; that has been key to so much of what I have done and how I have lived. Dad, I find more and more that I share many natural similarities to you; and I find more and more that this is a good thing. Justin, whether you know it or not, you are much of the reason that I ever cared about doing well in school. Though I know you would contest it, I am convinced to this day that you are far more intelligent than I. Micah, Noah, and Samuel, you influenced me less in these developmental ways since you're younger than I; but you deserve recognition for putting up with me all these years. Rachel, that you are choosing to become part of my family is astounding to me. I look forward to our life together.

ACKNOWLEDGMENTS

As with any accomplishment, there are many to be acknowledged for their contributions (more or less directly) to this work. I would be remiss if I did not first acknowledge God (1) for providing me with any abilities I do have and (2) for filling in the gaps with support from friends, family, and advisors. His grace and generosity to me are unfathomable.

During my somewhat extended tenure as a graduate student, many people served as advisors to me. Judy Walker provided me with much encouragement, direction, and patience as my first advisor. Her support during the early years of my graduate school experience was invaluable, and the time that we spent laughing during meetings helped to make the experience enjoyable. I am also grateful to Vladimir Itskov and Carina Curto who took something of a risk in allowing me to work with them later on. Vladimir taught me that I need to be willing to fight for any worthwhile idea or approach, and Carina kept me encouraged as I fought some of these fights. Vladimir pushed me to try to keep up as he leaped ahead in his thoughts, and Carina demonstrated that even complex ideas can be crafted and organized in such a way as to make them seem straightforward. Lastly, in inviting me to visit Janelia Research Campus, Eva Pastalkova assumed the role of an advisor of sorts. Although Eva never served as a mathematics advisor, she spent much time advising me as I found myself floundering to stay afloat in the foreign world of neuroscience. It was encouraging to work with such a principled scientist as I found myself questioning so much of what science seems to have become, and it was good to share with Eva in conversations of deeper importance than science.

In addition to Vladimir, Carina, and Judy, I would also like to explicitly thank the remainder of my supervisory committee: Jamie Radcliffe and Khalid Sayood. Jamie

unhesitatingly agreed to serve as a reader on this committee at the last minute, and he did this despite having had to read my combinatorics "proofs" during my first semester of graduate school. Khalid always offered a much-appreciated perspective on work and life. Though not part of my committee, the math department staff made my life easier during this time. Marilyn Johnson and Liz Youroukos, in particular, helped me with various things throughout the course of graduate school; and each slipped me lots of, uh, brain food to help me along.

Those who have known me beyond the walls of academia during graduate school have also contributed much toward this accomplishment. During this time, these friendships have kept me sane and grounded. It is, perhaps, unreasonable to try to mention all of the people who have aided me with their friendship over these years; but that will not stop me from trying.

To those who started school with me in the Fall of 2007, I appreciated our time together learning of analysis, algebra, combinatorics, and the such; and, yes, I even appreciated your jabs at me for being mostly indistinguishable from Orlando Bloom. To my many officemates throughout the years, I appreciated our time together and the many conversations that you each allowed me to thrust upon you while I was avoiding real work. I wouldn't have expected my life in a math department to contain so many thoughtful conversations about theology.

Mark DeBoer, moving to Lincoln was easier because I knew we would be living together at first; I appreciated our years living together. David Lessor and Joe LaChance, eating pizza, playing Halo, and climbing with you guys kept me feeling mostly normal as I fell deeper into the world of math. Mike Janssen and Tanner Auch, it's somewhat shocking how much time you were willing to spend with me outside of the time spent together in both school and church; shared meals, board games, *King of the Hill*, and vidya games with you were staples of my time in Lincoln.

Laura Janssen and Amy Auch, you facilitated this time together and became great friends in the process. Mike Callen, thank you for opening up your house to me while I was still mostly a stranger; no other close friendship of mine has ever formed quite so quickly as ours.

My time in Lincoln was truly special in that my school and church communities overlapped heavily. To my Grace Chapel community, thank you for all of your generosity. During these years in Lincoln, my close friends the Janssens, Auchs, and Rawsons probably fed me more than I fed myself. The opportunity to share in life with the same people in school, in church, and beyond was a blessing that I didn't fully appreciate at the time. Grace Fellowship Church, you made being in State College during my last year of graduate school not just bearable but worthwhile.

So many other people should be mentioned for some reason or another: Ben Nolting, Brian Kloppenborg, David and Jenny Rawson, Doug Dailey, Warren Wright, Jon Walker, David Beevers, Brian and Gretchen Bredemeier, Yingxue Wang, and Brian Lustig, to name a few. To those I have not mentioned, I appreciated our friendships, too. From others in my joint math/church community to landlords to friendships continued from times past and to many others, thank you.

Contents

Chapter 1

Sequential information in the brain

1.1 Phenomenology

The inner workings of the brain are still poorly understood. Study it though we may, the deep truths of its inner workings—the ways that it both processes and stores information—elude us. Even in organisms far simpler than humans, this problem remains. In 1986, White et al. [1] revealed the the entire connectome—the neurons and their synaptic connections—for the hermaphroditic roundworm C. elegans. Yet with this map of all 302 neurons and around 7000 connections, we still do not fully understand how the brains of C. elegans work. Comparing this to the estimated 86 billion neurons in the average adult human nervous system [2], the task of understanding our own brains is daunting to say the least. As such, it is not surprising that the body of literature describing and modeling the observed phenomena of the brain keeps growing. Here, we develop tools for use in the analysis of neuronal data that accompanies these phenomena. Though our tools are of general purpose, we focus on specific phenomena observed in a particular area of the mammalian brain: the hippocampus.

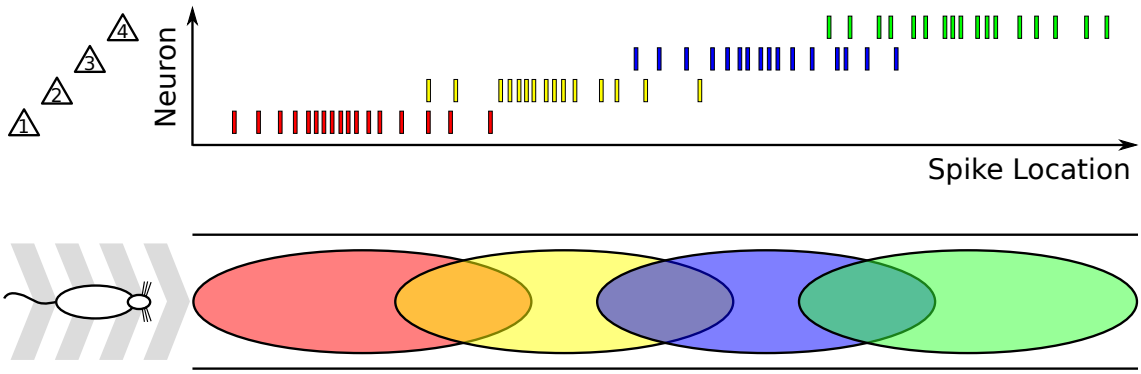


Figure 1.1: Place fields and neuronal spiking. The animal here has four place cells (not pictured) each with a preference for spiking while the animal is at a particular location within a straight track. The four place fields are shown as colored regions on the track (bottom). The traversal of such a track results in the generation of a neuronal sequence (top) called a place-field sequence, which reflects the direction of traversal through the track.

The hippocampus is widely believed to aid in two main functions: episodic memory and spatial navigation. Episodic memories are, broadly speaking, memories that are associated with events in one's life (times, places, emotions, etc.); this type of memory is in contrast with other types of memory such as semantic memory (knowledge of facts) and implicit memory (e.g., how to ride a bike). Evidence for the participation of the hippocampus in the formation of new episodic memories was first reported in [3] in 1957. A man referred to as Patient HM underwent a surgery to remove large portions of his hippocampal formation in an attempt to be cured of severe epileptic seizures. Although the surgery was successful in regards to reducing the frequency and severity of the seizures, there was an unexpected side effect: Patient HM no longer seemed to be capable of forming new memories about his experiences. Research on Patient HM and his condition continued until his death in 2008.

The clearest evidence of the participation of the hippocampus in spatial navigation is the existence of place cells, first observed in 1971 in [4]. These neurons have a preference for spiking when the animal is occupying a certain location in space, as

illustrated in Figure 1.1. The spatial area in which a place cell has a tendency to spike is called its place field. Traversal of a series of place fields results in the generation of a neuronal sequence called a place-field sequence, such as pictured in Figure 1.1. It is believed (e.g., [5], [6], [7]) that generation of place-field sequences is intimately related to, and perhaps dependent upon, a hippocampal oscillation of the local-field potential (LFP) called the theta oscillation, which exists whenever an animal is in locomotion.

A hippocampal event called a sharp-wave ripple (SWR) has also been tied to the formation of new episodic memories. Such an event is illustrated in Figure 1.2. As pictured in this figure, a SWR is characterized by two main features: (1) a sharp wave, which is a separation in the local-field potential (LFP) across the pyramidal layer of the hippocampus, and (2) a burst in neuronal activity, often observed as a high-frequency ripple in the LFP within the pyramidal layer. When SWRs were selectively eliminated from neural activity during post-training memory-consolidation periods in [8], animals exhibited reduced performance in spatial memory tasks, indicating that SWRs are necessary for the consolidation of episodic memories.

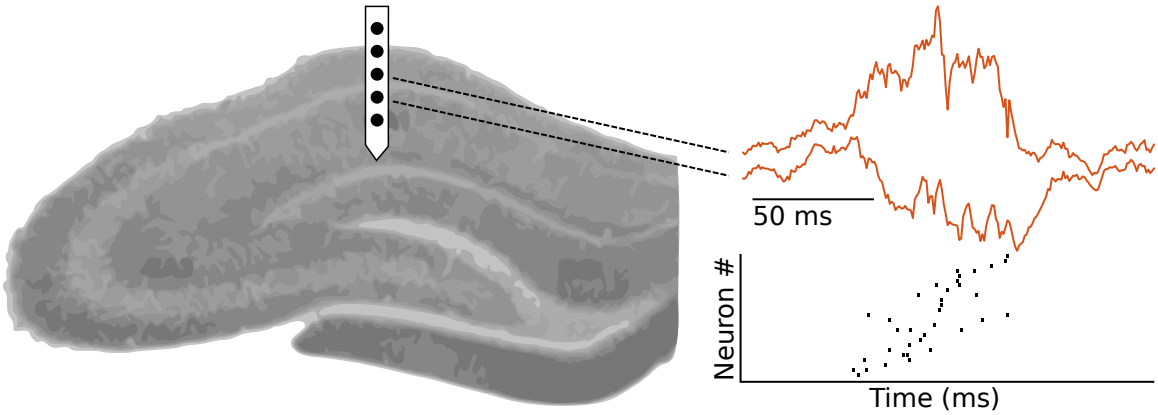
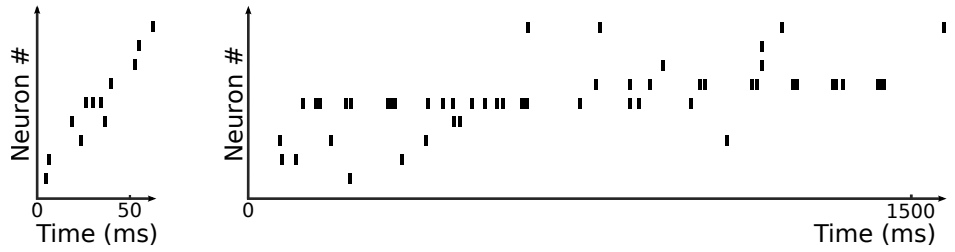


Figure 1.2: A sharp-wave ripple in the hippocampus. On the left, we see an image of a rat hippocampus with a probe across the pyramidal layer of the CA1 region of the hippocampus. In the upper-right section of this figure, we see a real example of local-field potentials from above and below the pyramidal layer. This separation of the local-field potential is called a sharp wave. Below the sharp wave, the increase in spiking that accompanies the SWR is showin in a spike raster plot.

1.2 Neuronal sequences

Given that neuronal sequences are important for storage and processing of information in the brain, it is crucial that we possess a reasonable model of sequences. Similarities between place-field sequences and SWR sequences [9] give reason to believe that properties such as sequence duration and total number of spikes (and even the precise timing of spikes) are perhaps less important (or, rather, manifestations of) some other properties that are less immediately obvious when considering and comparing sequences. An appropriate sequence model should ideally capture these intrinsic qualities and, in so doing, address the fundamental question of what a sequence actually is.

With clean illustrations such as that given in Figure 1.1, the concept of sequence seems simple: A sequence is intuitively, more or less, just an ordering of neurons. Though other characteristics stand out, the fact that there seems to be a clear ordering



(a) SWR (b) Place-field

Figure 1.3: Sequences from rat hippocampus. The SWR sequence in panel (a) and the place-field sequence in panel (b) are very different from each other in many ways including duration and number of spikes. Still, these sequences share a similar trend in the general order in which neurons seem to spike.

is a large part of what makes a sequence seem to be structured to us. This intuition, however, breaks down as we start to look noisy, real-world data such as that in Figure 1.3. How do the two sequences in this figure compare to each other? These sequences appear to the eye to have similar trends in their neuronal firing patterns. How can we characterize and quantify this relationship? This example immediately reveals that our intuitive notion of a sequence is not broad enough to encapsulate the sequences observed in real data. The situation becomes even hairier when we think about asking a computer to tell us which sequences are similar.

In the literature that deals with neuronal sequences, oversimiplified mathematical models have arisen to describe the concept of a sequence. These models have served a purpose and pushed the field forward, but they are limited in their representation of such a broad class of events. Chapter 2 introduces a generalized model that more robustly represents these sequences.

1.3 Summary of thesis

In the first half of this thesis (Chapters 2–3), we introduce a novel tool for characterizing and comparing neuronal sequences: the bias matrix. Chapter 2 provides

characterizations of neuronal sequences, including the introduction of the bias matrix (Section 2.3). In Chapter 3, we investigate mathematical properties of bias matrices, including a description of their relationship to the previously used center-of-mass representation (Section 3.1), a proof that all bias matrices lie in the fundamental polytope (Theorem 3.25 in Section 3.3), and a proof that the adjacency matrix of any simple digraph can be realized as the sign matrix of some bias matrix (Theorem 3.34 in Section 3.4).

The second part of this thesis (Chapters 4–5) describes how we use these tools to analyze neuronal sequences from the hippocampus of rats. Chapter 4 uses the bias matrix to introduce a statistical tool for detecting non-random correlations among sequences of neuronal activation. Chapter 5 introduces the experiment in which the data was recorded and describes the various types of sequences that are of interest. In Section 5.3, the techniques developed in Chapter 4 are used to investigate the similarities between the aforementioned sequence types. Appendix A serves as a README file for the code base that was developed for these analyses. Most of the code used for this analysis was written from scratch for this analysis.

Chapter 2

Characterizations of sequences

As discussed in Section 1.2, representation of neuronal sequences as orderings of neurons is appealing because it makes sequences easy to describe and easy to interpret; but it falls short in that (1) it does not capture the complex relationships that exist between neurons, especially as observed in noisy data, and (2) it has no simple, realistic biological interpretation. In this chapter, we describe a new representation of sequences that addresses these issues and, as we will see in Chapter 3, generalizes the concept of a neuronal ordering.

2.1 A combinatorial characterization

Throughout the remainder of this document, we denote the collection of neurons by $\mathcal{N} := [n]$, where $n \in \mathbb{N}$. The raw form of a sequence with which we start is called a spike train:

Definition 2.1. A *spike train* is a set of pairs of firing times and neurons, that is, a set of the form $\{(t, i) \mid t \in \mathbb{R},\ i \in \mathcal{N}\}$.

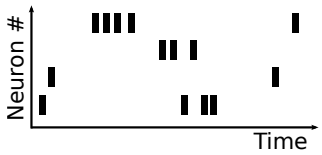


Figure 2.1: Spike raster plot. To get a sequence from a spike train pictured as a spike-raster plot (as shown here), we simply list the neuron for each spike in order of occurrence (left to right). Hence, the corresponding sequence is $\mathbf{s} = (1, 2, 4, 4, 4, 4, 3, 3, 1, 3, 1, 1, 2, 4) = 12444433131124$.

We have already seen graphical examples of spike trains in Figures 1.2 and 1.3 in the form of spike raster plots. As seen in the real-world sequences introduced in Chapter 1, neuronal sequences occur on various timescales; but their sequential nature is independent of the timescales on which they occur. As such, an appropriate sequence representation will be timescale invariant while still maintaining complex relationships between the spikes of neurons. Although spike trains maintain complex relationships between the spikes of neurons, this representation is not timescale-invariant since each spike has an associated time. If we instead discard this time information and maintain only the order in which spikes occur, the resultant representation will be timescale-invariant and also represent complex inter-spike relationships. This brings us to our formal definition of a sequence.

Definition 2.2. A *sequence* is a finite list of neurons.

If $\mathbf{s} = (s_1, s_2, \ldots, s_\ell)$ is a sequence, we will often write $\mathbf{s}$ as $\mathbf{s} = s_1 s_2 \cdots s_\ell$. Given a spike train $T = \{(t_1, i_1), \ldots, (t_\ell, i_\ell)\}$ with $t_1 \leq \cdots \leq t_\ell$ and $i_k < i_{k+1}$ if $t_k = t_{k+1}$, the corresponding sequence is $(i_1, \ldots, i_\ell)$. An example of this process is illustrated in Figure 2.1.

The set of all sequences on the neuron set $\mathcal{N}$ is represented by $\mathcal{S}(\mathcal{N})$ or, if $\mathcal{N}$ is understood, by $\mathcal{S}$. Table 2.1 contains some terminology and notation for working with a sequence $\mathbf{s} = (s_1, \ldots, s_\ell)$ and a neuron i.

Table 2.1: The terminology of sequences

Notation	Terminology	Meaning
	a *spike* of $\mathbf{s}$	an entry of a sequence $\mathbf{s}$
	i *spikes* in $\mathbf{s}$	$i = s_k$ for some $k \in [\ell]$
$\text{len}(\mathbf{s})$	*length* of $\mathbf{s}$	the number of spikes in $\mathbf{s}$
$\text{supp}(\mathbf{s})$	*support* of $\mathbf{s}$	collection of neurons that spike in $\mathbf{s}$
$L_i(\mathbf{s})$	*spike set* of i	collection of indices where i spikes
$c_i(\mathbf{s})$	*spike count* of i	number of times that i spikes
$\rho_i(\mathbf{s})$	*firing rate* of i	proportion of spikes that are equal to i

Example 2.3. With $\mathbf{s}$ from the previous example, we see that neuron 1 spikes four times and has spike set $L_1(\mathbf{s}) = \{1, 9, 11, 12\}$; hence, the spike count of 1 in $\mathbf{s}$ is $c_1(\mathbf{s}) = 4$. Similarly, $c_2(\mathbf{s}) = 2$, $c_3(\mathbf{s}) = 3$, $c_4(\mathbf{s}) = 5$, and $c_i(\mathbf{s}) = 0$ for any other neuron $i \in \mathcal{N}$. Since $\text{len}(\mathbf{s}) = 14$, the firing rate of each neuron in $\mathbf{s}$ is $\rho_1(\mathbf{s}) = \frac{4}{14}$, $\rho_2(\mathbf{s}) = \frac{2}{14}$, $\rho_3(\mathbf{s}) = \frac{3}{14}$, and $\rho_4(\mathbf{s}) = \frac{5}{14}$.

Definition 2.4. Given a sequence $\mathbf{s} = (s_1, \ldots, s_\ell)$ and an index set $K = \{k_1, \ldots, k_{\ell'}\} \subseteq [\ell]$ with $k_1 < \cdots < k_{\ell'}$, we define the *subsequence* of $\mathbf{s}$ corresponding to K to be $\mathbf{s}_K := (s_{k_1}, \ldots, s_{k_{\ell'}})$. We say that a sequence $\mathbf{s}'$ is a subsequence of $\mathbf{s}$ if $\mathbf{s}' = \mathbf{s}_K$ for some index set $K \subseteq [\ell]$.

Example 2.5. Let $\mathbf{s} = (1, 2, 4, 4, 4, 4, 3, 3, 1, 3, 1, 1, 2, 4)$ be the sequence from Figure 2.1, and let $K_1 = [6]$ and $K_2 = \{1, 2, 9, 11, 12, 13\}$. Then the subsequences of $\mathbf{s}$ corresponding to K_1 and K_2 are $\mathbf{s}_{K_1} = (1, 2, 4, 4, 4, 4)$ and $\mathbf{s}_{K_2} = (1, 2, 1, 1, 1, 2)$, respectively.

The following operations will be useful in our discussion of sequences.

Definition 2.6. Given any two sequences $\mathbf{u} = (u_1, \ldots, u_\ell)$ and $\mathbf{v} = (v_1, \ldots, v_m)$, define the *concatenation* of $\mathbf{u}$ with $\mathbf{v}$ to be the sequence $\mathbf{u}\cdot\mathbf{v} = \mathbf{uv} = (u_1, \ldots, u_\ell, v_1, \ldots, v_m)$.

Notation 2.7. For any neuron $i \in \mathcal{N}$ and any positive integer $m \in \mathbb{N}$, let i^m denote the sequence $(s_1, \ldots, s_m)$ with $s_k = i$ for all $k \in [m]$.

2.2 Center-of-mass sequences

A sequence can induce an ordering by many means. One sandard way of performing this reduction is by considering the average location in which a neuron fires.

Definition 2.8. For each neuron $i \in \mathcal{N}$ and sequence $\mathbf{s} = (s_1, \ldots, s_\ell)$, we define the *(unnormalized) center of mass* of neuron i in sequence $\mathbf{s}$ to be

$$\operatorname{center}_i(\mathbf{s}) := \begin{cases} \frac{1}{c_i(\mathbf{s})} \sum_{k \in L_i(\mathbf{s})} k, & c_i(\mathbf{s}) > 0; \\ \frac{\operatorname{len}(\mathbf{s})+1}{2}, & c_i(\mathbf{s}) = 0. \end{cases}$$

Note that $\operatorname{center}_i(\mathbf{s}) \in [1, \operatorname{len}(\mathbf{s})]$ for all $i \in \mathcal{N}$. Moreover, when a neuron does not fire in a sequence, the corresponding center-of-mass value is chosen so that it falls in the middle of the range $[1, \ell]$. Thus, $\operatorname{center}_i(\mathbf{s})$ reduces all spikes from neuron i in sequence $\mathbf{s}$ to a single value; but this value is dependent on the length of $\mathbf{s}$. To allow for comparison of sequences of different lengths, we shift and normalize $\operatorname{center}_i(\mathbf{s})$ so that the resultant value lies in the interval $(-1, 1)$ regardless of the length of $\mathbf{s}$:

Definition 2.9. For each neuron $i \in \mathcal{N}$ and sequence $\mathbf{s} = (s_1, \ldots, s_\ell)$, we define the *(normalized) center of mass* of neuron i in sequence $\mathbf{s}$ to be

$$\gamma_i(\mathbf{s}) := \frac{2 \operatorname{center}_i(\mathbf{s}) - 1}{\operatorname{len}(\mathbf{s})} - 1.$$

Note that with this definition, neurons that do not fire in the sequence have $\gamma_i(\mathbf{s}) = 0$. If the sequence consists of just one spike, then the corresponding neuron has $\operatorname{center}_i(\mathbf{s}) = 1$ and hence $\gamma_i(\mathbf{s}) = 0$. On the other hand, if a neuron fires only once in the sequence and this corresponds to the first or last spike, then $\gamma_i(\mathbf{s}) = -1 + \frac{1}{\operatorname{len}(\mathbf{s})}$ or $\gamma_i(\mathbf{s}) = 1 - \frac{1}{\operatorname{len}(\mathbf{s})}$, respectively, so that the endpoints of the $[-1, 1]$ range are only

attained in the limit as $\text{len}(\mathbf{s}) \to \infty$. These centers of mass can now be used to induce an ordering on the underlying neurons.

Definition 2.10. We define the *center-of-mass ordering* of $\mathbf{s}$ to be $\sigma(\mathbf{s}) := (i_1, \ldots, i_n)$ where $\{i_1, \ldots, i_n\} = [n]$ are such that $\gamma_{i_1}(\mathbf{s}) \leq \cdots \leq \gamma_{i_n}(\mathbf{s})$ and, in the case that $\gamma_{i_k}(\mathbf{s}) = \gamma_{i_{k+1}}(\mathbf{s})$, such that $i_k < i_{k+1}$.

Example 2.11. Let $\mathbf{s} = 12444433131124$. Then, as we saw in Example 2.3, $L_1(\mathbf{s}) = \{1, 9, 11, 12\}$ and $c_1(\mathbf{s}) = 4$. Now, $\text{center}_1(\mathbf{s}) = \frac{1}{c_1(\mathbf{s})} \sum L_1(\mathbf{s}) = \frac{33}{4}$ and $\gamma_1(\mathbf{s}) = \frac{2(33/4)-1}{14} - 1 = \frac{3}{28}$. Similarly, we find that $\text{center}(\mathbf{s}) = (\frac{33}{4}, \frac{15}{2}, \frac{25}{3}, \frac{32}{5})$ and $\gamma(\mathbf{s}) = (\frac{3}{28}, 0, \frac{5}{42}, -\frac{11}{70})$. Further, $\gamma_4(\mathbf{s}) < \gamma_2(\mathbf{s}) < \gamma_1(\mathbf{s}) < \gamma_3(\mathbf{s})$ and, hence, $\sigma(\mathbf{s}) = (4, 2, 1, 3)$.

It is worth mentioning that when the concept of centers of mass is used in the literature, the starting point is usually spike trains, not our concept of a neuronal sequence. When starting with spike trains, the spike times of a neuron are averaged instead of its indices in the sequence; these average firing times then induce an ordering in the same way that the centers of mass do.

2.3 Pairwise biases

If each neuron active in a sequence spikes only once, then the sequence itself is an ordering of its active neurons. In such a case, the center-of-mass ordering equals the original sequence. A broader class of sequences induce an ordering in a similarly unambiguous way. For instance, in the sequence 3311222, every spike of neuron 3 comes before every spike of neuron 1; further, every spike of neuron 1 occurs before every spike of neuron 2. Hence, 3311222 induces the unambiguous order 312. Although this ordering can be computed with centers of mass, the ordering is also immediately

obvious upon inspection of the sequence since all of the spikes from each neuron are grouped together. We give a name to sequences such as this.

Definition 2.12. Disjoint sets $I, J \subseteq \mathbb{N}$ are *interlaced* if there exist $i, i' \in I$ and $j, j' \in J$ such that $i < j$ and $i' > j'$.

Definition 2.13. A sequence $\mathbf{s}$ is *pure* if the spike sets of the active neurons are not interlaced.

In a pure sequence, all of the spikes from one neuron must follow (or precede) all of the spikes from any other neuron. In particular, the indices of the spikes of each individual neuron form a contiguous set of natural numbers. To get a sequence that is not pure from one that is pure, we could transpose adjacent spikes from different neurons. For example, the pure sequence 111122223333 can be transformed into the impure sequence 111212223333 by transposing the fourth and fifth spikes in the pure sequence. In a sense, this new sequence is minimally impure: The spikes from neuron 3 are still contiguous, and only one pair of spikes from neurons 1 and 2 are "out of order." This perspective suggests that the purity of a sequence (1) is somehow related to the pairwise relationships between the spikes of different neurons and (2) is dependent on the collection of all such pairs of neurons. In other words, the purity of a sequence depends on the bias of each pair of neurons to fire in a particular order. We begin to quantify these biases with the bias-count matrix below.

Definition 2.14. The *bias-count matrix* of the sequence $\mathbf{s}$ is the matrix $C(\mathbf{s}) := [c_{ij}(\mathbf{s})]_{i,j\in\mathcal{N}}$ where $c_{ij}(\mathbf{s}) := |L_{ij}(\mathbf{s})|$ and $L_{ij}(\mathbf{s}) := \{(k, k') \mid k < k',\ s_k = i,\ \text{and}\ s_{k'} = j\}$. In words, $c_{ij}(\mathbf{s})$ is the number of length-2 subsequences of $\mathbf{s}$ that are equal to (i, j).

Example 2.15. Consider the sequence $\mathbf{s} = (1, 2, 4, 4, 4, 4, 3, 3, 1, 3, 1, 1, 2, 4)$ from Fig-

ure 2.1. Note that $L_1(\mathbf{s}) = \{1, 9, 11, 12\}$ and $L_2(\mathbf{s}) = \{2, 13\}$. Hence, we see that

$$L_{12}(\mathbf{s}) = \{(1,2), (1,13), (9,13), (11,13), (12,13)\}$$

and

$$L_{21}(\mathbf{s}) = \{(2,9), (2,11), (2,12)\};$$

so $c_{12}(\mathbf{s}) = |L_{12}(\mathbf{s})| = 5$ and $c_{21}(\mathbf{s}) = |L_{21}(\mathbf{s})| = 3$. In a similar way for the other entries $c_{ij}(\mathbf{s})$, we find that the bias-count matrix for $\mathbf{s}$ is

$$C(\mathbf{s}) = \begin{bmatrix} 6 & 5 & 4 & 8 \\ 3 & 1 & 3 & 6 \\ 8 & 3 & 3 & 3 \\ 12 & 4 & 12 & 10 \end{bmatrix}.$$

A few simple facts about the bias-count matrix are immediate. First, since the definition of the bias-count matrix arose out of a discussion of pure sequences, it is worth noting that the bias-count matrix completely captures whether or not a sequence is pure.

Lemma 2.16. *A sequence* $\mathbf{s}$ *with* $\operatorname{supp}(\mathbf{s}) = \mathcal{N}$ *has at least* $\binom{|\mathcal{N}|}{2}$ *non-zero, off-diagonal entries, with equality if and only if* $\mathbf{s}$ *is pure.*

Proof. Any $C(\mathbf{s})$ must obviously have at least $\binom{|\mathcal{N}|}{2}$ non-zero, off-diagonal entries since each of the $\binom{|\mathcal{N}|}{2}$ pairs $i, j \in \mathcal{N}$ of active neurons must fire in at least one of the two possible orderings (i, j) or (j, i). If $\mathbf{s}$ is pure, it must be the case that exactly one of the orderings (i, j) or (j, i) appears in $\mathbf{s}$ for each $i, j \in \mathcal{N}$ with $i \neq j$, of which there are exactly $\binom{|\mathcal{N}|}{2}$. If $C(\mathbf{s})$ has more than $\binom{|\mathcal{N}|}{2}$ non-zero, off-diagonal entries, then, by the pigeonhole principle, there exist distinct neurons $i, j \in \mathcal{N}$ such that $c_{ij}(\mathbf{s}) > 0$

and $c_{ji}(\mathbf{s}) > 0$; so we must have that the spikes of i and j are interlaced, meaning that $\mathbf{s}$ is not pure. This finishes the proof. □

Next, one cannot always recover a sequence from its bias-count matrix. Consider, for example, $\mathbf{s}_1 = 1221$ and $\mathbf{s}_2 = 2112$. Then $C(\mathbf{s}_1) = C(\mathbf{s}_2) = \left[\begin{smallmatrix} 1 & 2 \\ 2 & 1 \end{smallmatrix}\right]$. Hence, thought of as a map, $C : \mathcal{S} \to \mathbb{N}^{|\mathcal{N}|\times|\mathcal{N}|}$ is not injective. Though we cannot recover $\mathbf{s}$ from $C(\mathbf{s})$, we can recover the spike counts $c_i(\mathbf{s})$. To see this, first note that

$$c_{ii}(\mathbf{s}) = \binom{c_i(\mathbf{s})}{2}.$$

Since $f : \mathbb{N} \to \mathbb{N}$ by $f(n) = \binom{n}{2}$ is injective, it has an inverse; therefore,

$$c_i(\mathbf{s}) = f^{-1}(c_{ii}(\mathbf{s})) = \frac{1}{2} + \frac{1}{2}\sqrt{1 + 8c_{ii}(\mathbf{s})}.$$

Last, note that $c_{ij} + c_{ji} = c_i c_j$. In words, $c_{ij} + c_{ji}$ is equal to the product of the number of spikes of i and of j. This is easy to see because we just choose one index for i and one index for j (which can be done in $c_i c_j$ ways); in each such choice, either i follows j (counted by c_{ji}) or j follows i (counted by c_{ji}).

Though $C(\mathbf{s})$ encapsulates the idea of purity, it does so in a somewhat unsatisfactory way. The sequences 123 and 112233, for instance, are both pure sequences with the same neuronal ordering, yet their bias-count matrices

$$C(123) = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{and} \qquad C(112233) = \begin{bmatrix} 1 & 4 & 4 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix}$$

are quite different. The issue is that 112233 has more pairs of spikes than 123 does. We can correct for this by normalizing each entry by an appropriate value. How likely is neuron j to follow neuron i? There are c_{ij} index pairs in which j follows i out of a

total of $c_i c_j$ possible index pairs. This leads us to the definition of a second type of bias matrix.

Definition 2.17. The *probability-bias matrix* of a sequence $\mathbf{s}$ is the matrix $B_{\text{prob}}(\mathbf{s}) := [b_{ij}(\mathbf{s})]_{i,j\in\mathcal{N}}$ where $b_{ij}(\mathbf{s}) := \frac{c_{ij}(\mathbf{s})}{c_i(\mathbf{s})c_j(\mathbf{s})}$.

Example 2.18. The probability-bias matrix for the sequence $\mathbf{s} = 12444433131124$ from Figure 2.1 is

$$B_{\text{prob}}(\mathbf{s}) = \begin{bmatrix} 6/16 & 5/8 & 4/12 & 8/20 \\ 3/8 & 1/4 & 3/6 & 6/10 \\ 8/12 & 3/6 & 3/9 & 3/15 \\ 12/20 & 4/10 & 12/15 & 10/25 \end{bmatrix} = \begin{bmatrix} 3/8 & 5/8 & 1/3 & 2/5 \\ 3/8 & 1/4 & 1/2 & 3/5 \\ 2/3 & 1/2 & 1/3 & 1/5 \\ 3/5 & 2/5 & 4/5 & 2/5 \end{bmatrix}.$$

The probability-bias matrix maps the set $\mathcal{S}$ of sequences into the matrices $[0,1]^{|\mathcal{N}|\times|\mathcal{N}|}$, and it does so in a way that each off-diagonal entry of the matrix has a natural interpretation in terms of how likely it is to see a pair of spikes in a given order. As such, it should hold that $b_{ij} + b_{ji} = 1$. Recalling that $c_{ij} + c_{ji} = c_i c_j$, we see that this indeed holds:

$$b_{ij} + b_{ji} = \frac{c_{ij}}{c_i c_j} + \frac{c_{ji}}{c_j c_i} = \frac{c_{ij} + c_{ji}}{c_i c_j} = \frac{c_i c_j}{c_i c_j} = 1.$$

Stated simply, when i and j both spike in a sequence, each pair of spikes must occur in one order or the other. As defined, this likelihood is global in terms of the sequence, which gives it a sort of skew-symmetric nature: $b_{ij} = 1 - b_{ji}$. As such, it is clear that not every point in the hypercube $[0,1]^{|\mathcal{N}|\times|\mathcal{N}|}$ can be obtained as a probability-bias matrix. Though that much is known simply because no irrational coordinate can ever be hit by the map, the above relation shows that at least one further relation is imposed on this image. In a sense, this relation suggests that the bias matrix is still not quite in the right form: Since $b_{ij}+b_{ji} = 1$, we see that $b_{ij} - \frac{1}{2} = \frac{1}{2} - b_{ji} = -(b_{ji} - \frac{1}{2})$. Because the relation comes from the fact that the probability biases are probabilities,

the entries are centered at $\frac{1}{2}$, whereas we might more naturally think of a non-bias (i.e., a maximally unbiased bias) as being equal to 0. Similarly, we might want the sign of a bias to convey the tendency of neurons to spike in the given order (a positive bias) or the opposite order (a negative bias). Perhaps we then care more about the difference between b_{ij} and b_{ji} than we do about each of them individually. This leads us to define our final version of the bias matrix.

Definition 2.19. The *skew-bias matrix* of the sequence $\mathbf{s}$ is the matrix $B_{\text{skew}}(\mathbf{s}) := B_{\text{prob}}(\mathbf{s}) = B_{\text{prob}}(\mathbf{s})^T = [\beta_{ij}(\mathbf{s})]_{i,j\in\mathcal{N}}$ with entries $\beta_{ij}(\mathbf{s}) = b_{ij}(\mathbf{s}) - b_{ji}(\mathbf{s}) = \frac{c_{ij}(\mathbf{s})-c_{ji}(\mathbf{s})}{c_i(\mathbf{s})c_j(\mathbf{s})}$. The skew-bias matrix $B_{\text{skew}}(\mathbf{s})$ is a scaling of the skew-symmetric part of the probability-bias matrix $B_{\text{prob}}(\mathbf{s})$.

Note that when $i \neq j$, $\beta_{ij}(\mathbf{s}) = b_{ij} - b_{ji} = b_{ij} - (1 - b_{ij}) = 2b_{ij} - 1$. The skew biases are now centered at zero and have a sign that indicates the direction of the bias. Further, the skew-bias matrix is skew-symmetric as a matrix since $\beta_{ij} = b_{ij} - b_{ji} = -(b_{ji} - b_{ij}) = -\beta_{ji}$.

Example 2.20. The skew-bias matrix for the sequence $\mathbf{s} = (1, 2, 4, 4, 4, 4, 3, 3, 1, 3, 1, 1, 2, 4)$ from Figure 2.1 is

$$B_{\text{skew}}(\mathbf{s}) = \begin{bmatrix} 0 & 1/4 & -1/3 & -1/5 \\ -1/4 & 0 & 0 & -1/5 \\ 1/3 & 0 & 0 & -3/5 \\ 1/5 & 1/5 & 3/5 & 0 \end{bmatrix}.$$

Here, it is obvious at a glance that, for instance, neuron 2 tends to follow neuron 1 and that neurons 2 and 3 are maximally unbiased toward each other.

2.4 Higher-order biases

When creating a mathematical model, it is important to question what information the model discards. The pairwise biases from the previous section attempt to capture the tendency of pairs of neurons to fire in a particular order. While this certainly preserves some information about complex relationships between neurons/spikes, it may at times discard more information than is desired.

Example 2.21. Consider the sequences $\mathbf{s}_1 = 123321$ and $\mathbf{s}_2 = 321123$. These sequences are indistinguishable by pairwise biases:

$$C(\mathbf{s}_1) = C(\mathbf{s}_2) = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}.$$

Note that, since the probability- and skew-bias matrices are computed from the bias-count matrix, these matrices also cannot distinguish between $\mathbf{s}_1$ and $\mathbf{s}_2$.

We might want, however, for the sequences from the above example to be distinguishable using some bias-like method. This may or may not be possible for all pairs of sequences, but we can extend the idea behind the pairwise-bias matrices to consider more neurons within each individual bias. For instance, we can count the subsequences that are equal to each permutation of 123 (where, before, we counted the subsequences equal to each permutation of 12). We call these higher-order biases.

Definition 2.22. For a list $\mathbf{v} = (v_1, \ldots, v_k)$ of k distinct neurons, we define

$$L_{\mathbf{v}}(\mathbf{s}) := \{(i_1, \ldots, i_k) \mid i_1 < \cdots < i_k \text{ and } s_{i_j} = v_j \text{ for all } j \in [k]\}.$$

to be the collection of subsequences of $\mathbf{s}$ that are equal to $\mathbf{v}$ and define

$$c_{\mathbf{v}}(\mathbf{s}) := |L_{\mathbf{v}}(\mathbf{s})|$$

to be the count of those subsequences; we call $c_{\mathbf{v}}(\mathbf{s})$ an *order-k bias count*. Moreover, analogously to the bias-count matrix, we define the *order-k bias-count vector*

$$C_k(\mathbf{s}) := \begin{bmatrix} c_{\mathbf{v}_1}(\mathbf{s}) \\ \vdots \\ c_{\mathbf{v}_{k'}}(\mathbf{s}) \end{bmatrix}$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_{k'}$ is the list of length-k biases in lexicographical order and $k' = \binom{|\mathcal{N}|}{k} k! = \frac{|\mathcal{N}|!}{(|\mathcal{N}|-k)!} = |\mathcal{N}|_{(k)}$.

Example 2.23. Continuing the previous example, let $\mathbf{s}_1 = 123321$ and $\mathbf{s}_2 = 321123$. Note that $L_{231}(\mathbf{s}_1) = \{(2,3,6),(2,4,6)\}$ and $L_{231}(\mathbf{s}_2) = \emptyset$. In particular, $c_{231}(\mathbf{s}_1) = 2$ but $c_{231}(\mathbf{s}_2) = 0$. Hence, the order-3 biases can distinguish between the sequences $\mathbf{s}_1$ and $\mathbf{s}_2$.

Note that $c_{\mathbf{v}}$ is a generalization of c_{ij} and that $c_{ij} = c_{(i,j)} = c_{\mathbf{v}}$ for $\mathbf{v} = (1,2)$. This also unifies the definitions of the spike-count c_i and the bias-count c_{ij}; that is, c_i and c_{ij} are order-1 and order-2 biases, respectively. But this added information comes at a cost: The number of order-k biases is $O(|\mathcal{N}|^k)$. With this increase in the number of biases, it is valuable to see that a higher-order bias vector contains at least as much information about a sequence as a lower-order conterpart.

Lemma 2.24. *The collection of order-$(k-1)$ biases can be computed from the collection of order-k biases if the spike counts are also known.*

Proof. Let $\mathbf{s} \in \mathcal{S}$ be given, and let $\mathbf{v} = (v_1, \ldots, v_{k-1}) \in \mathcal{N}^{k-1}$ be a list of distinct neurons. We want to compute the order-$(k-1)$ bias $c_{\mathbf{v}}(\mathbf{s})$ from the collection of order-

k biases. Recall that $c_{\mathbf{v}}(\mathbf{s})$ is the number of times that $\mathbf{v}$ appears as a subsequence of $\mathbf{s}$. Let U be the collection of length-k sequences of distinct neurons and of which $\mathbf{v}$ is a subsequence. To make this more precise, define the sets $V = \{v_1, \ldots, v_{k-1}\}$ and $I = \mathcal{N} \setminus V$. We can now write U as

$$U = \left\{ \mathbf{v}_{[m]} \cdot v \cdot \mathbf{v}_{\{m+1,\ldots,k\}} \mid v \in N,\ m \in [k] \right\}.$$

We want to double-count the number d of subsequences of $\mathbf{s}$ that are equal to some element of U. First, by definition of d and the bias count $c_{\mathbf{u}}(\mathbf{s})$, we have that $d = \sum_{\mathbf{u} \in U} c_{\mathbf{u}}(\mathbf{s})$.

To count in another way, let $B = \bigcup_{v \in N} L_v(\mathbf{s})$ be the collection of indices of $\mathbf{s}$ that are not spikes from the set V. Now, each subsequence of $\mathbf{s}$ that is equal to an element of U has the form $\mathbf{s}_{J \cup \{m\}}$ for some $m \in B$ and some $J \subseteq [\ell]$ with $\mathbf{s}_J = \mathbf{v}$; moreover, each such J and m yields such a subsequence. As there are $c_{\mathbf{v}}(\mathbf{s})$ such J and $\sum_{v \in N} c_v(\mathbf{s})$ such m, we find that $d = c_{\mathbf{v}}(\mathbf{s}) \sum_{v \in N} c_v(\mathbf{s})$. Finally, putting together the two expressions for d, we find that

$$c_{\mathbf{v}}(\mathbf{s}) = \left(\sum_{\mathbf{u} \in U} c_{\mathbf{u}}(\mathbf{s}) \right) \Big/ \left(\sum_{v \in N} c_v(\mathbf{s}) \right).$$

This completes the proof. □

Similarly to the pairwise biases, we may not care about subsequence counts $c_{\mathbf{v}}$ so much as the likelihood that the neurons $v_1, \ldots, v_k$ will appear in the order $\mathbf{v} = (v_1, \ldots, v_k)$ provided that they all appear.

Definition 2.25. Given a list $\mathbf{v} = (v_1, \ldots, v_k)$ of distinct neurons, define the *order-k probability-bias* to be the number $b_{\mathbf{v}}(\mathbf{s}) := \frac{c_{\mathbf{v}}(\mathbf{s})}{c_{v_1}(\mathbf{s}) \cdots c_{v_k}(\mathbf{s})}$. The *order-k probability-bias*

vector is the vector

$$B_k(\mathbf{s}) := \begin{bmatrix} b_{\mathbf{v}_1}(\mathbf{s}) \\ \vdots \\ b_{\mathbf{v}_{k'}}(\mathbf{s}) \end{bmatrix},$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_{k'}$ is the list of length-k biases in lexicographical order.

Note that the denominator of $b_{\mathbf{v}}(\mathbf{s})$ is the total number of length-k subsequences of $\mathbf{s}$ that contain all neurons $v_1, \ldots, v_k$; together with the fact that $c_{\mathbf{v}}(\mathbf{s})$ is a count of such sequences, we see that $b_{\mathbf{v}}(\mathbf{s})$ is really the proportion of all length-k subsequences of $\mathbf{s}$ containing $v_1, \ldots, v_k$ that are equal to $\mathbf{v}$.

Example 2.26. Let $\mathbf{s} = 12444433131124$. Then $L_1(\mathbf{s}) = \{1, 9, 11, 12\}$, $L_2(\mathbf{s}) = \{2, 13\}$, and $L_3(\mathbf{s}) = \{7, 8, 10\}$. So we see that $L_{123}(\mathbf{s}) = \{(1, 2, 7), (1, 2, 8), (1, 2, 10)\}$ and $L_{132}(\mathbf{s}) = \{(1, 7, 13), (1, 8, 13), (1, 10, 13), (9, 10, 13)\}$; hence,

$$c_{123}(\mathbf{s}) = |L_{123}(\mathbf{s})| = 3 \qquad \text{and} \qquad c_{132}(\mathbf{s}) = |L_{132}(\mathbf{s})| = 4.$$

Since $c_1(\mathbf{s}) = 4$, $c_2(\mathbf{s}) = 2$, and $c_3(\mathbf{s}) = 3$, we also have that

$$b_{123}(\mathbf{s}) = \frac{c_{123}(\mathbf{s})}{c_1(\mathbf{s})c_2(\mathbf{s})c_3(\mathbf{s})} = \frac{1}{8} \qquad \text{and} \qquad b_{123}(\mathbf{s}) = \frac{c_{123}(\mathbf{s})}{c_1(\mathbf{s})c_2(\mathbf{s})c_3(\mathbf{s})} = \frac{1}{6}.$$

Similarly, we can compute the remainder of the order-3 bias-count and probability-bias vectors. The vectors are shown as matrices here strictly for formatting purposes:

$$C_3(\mathbf{s}) = \begin{bmatrix} c_{123}(\mathbf{s}) & c_{213}(\mathbf{s}) & c_{312}(\mathbf{s}) & c_{412}(\mathbf{s}) \\ c_{124}(\mathbf{s}) & c_{214}(\mathbf{s}) & c_{314}(\mathbf{s}) & c_{413}(\mathbf{s}) \\ c_{132}(\mathbf{s}) & c_{231}(\mathbf{s}) & c_{321}(\mathbf{s}) & c_{421}(\mathbf{s}) \\ c_{134}(\mathbf{s}) & c_{234}(\mathbf{s}) & c_{324}(\mathbf{s}) & c_{423}(\mathbf{s}) \\ c_{142}(\mathbf{s}) & c_{241}(\mathbf{s}) & c_{341}(\mathbf{s}) & c_{431}(\mathbf{s}) \\ c_{143}(\mathbf{s}) & c_{243}(\mathbf{s}) & c_{342}(\mathbf{s}) & c_{432}(\mathbf{s}) \end{bmatrix} = \begin{bmatrix} 3 & 1 & 8 & 12 \\ 9 & 3 & 8 & 4 \\ 4 & 8 & 0 & 0 \\ 4 & 3 & 3 & 0 \\ 4 & 12 & 0 & 32 \\ 12 & 12 & 0 & 12 \end{bmatrix}$$

and

$$B_3(\mathbf{s}) = \begin{bmatrix} b_{123}(\mathbf{s}) & b_{213}(\mathbf{s}) & b_{312}(\mathbf{s}) & b_{412}(\mathbf{s}) \\ b_{124}(\mathbf{s}) & b_{214}(\mathbf{s}) & b_{314}(\mathbf{s}) & b_{413}(\mathbf{s}) \\ b_{132}(\mathbf{s}) & b_{231}(\mathbf{s}) & b_{321}(\mathbf{s}) & b_{421}(\mathbf{s}) \\ b_{134}(\mathbf{s}) & b_{234}(\mathbf{s}) & b_{324}(\mathbf{s}) & b_{423}(\mathbf{s}) \\ b_{142}(\mathbf{s}) & b_{241}(\mathbf{s}) & b_{341}(\mathbf{s}) & b_{431}(\mathbf{s}) \\ b_{143}(\mathbf{s}) & b_{243}(\mathbf{s}) & b_{342}(\mathbf{s}) & b_{432}(\mathbf{s}) \end{bmatrix} = \begin{bmatrix} 1/8 & 1/24 & 1/3 & 3/10 \\ 9/40 & 3/40 & 2/15 & 1/15 \\ 1/6 & 1/3 & 0 & 0 \\ 1/15 & 1/10 & 1/10 & 0 \\ 1/10 & 3/10 & 0 & 8/15 \\ 1/5 & 2/5 & 0 & 2/5 \end{bmatrix}.$$

We define no higher-order version of the skew-bias matrix B_{skew} because the skew-nature of B_{skew} arose from the fact that only two probabilities were involved for each pair of neurons, that is, from the fact that $b_{ij} + b_{ji} = 1$. In the higher-order version, a more complicated relationship exists, namely

$$\sum_{\sigma \in \text{Perms}(\{n_1,\ldots,n_k\})} b_{(n_{\sigma(i)},\ldots,n_{\sigma(n)})} = 1$$

for each set of distinct neurons $\{n_1, \ldots, n_k\} \subseteq \mathcal{N}$, where Perms(()$\Omega$) is the collection of permutations of the set Ω.

Chapter 3

Bias Matrices

In the previous chapter, we formalized the concept of a neuronal sequence and introduced the pairwise biases associated with those sequences. These pairwise biases—of which we have three separate formulations—capture the tendency of pairs of neurons to fire in a particular order. As the main drive of the discussion in the previous chapter was to motivate and define these biases, it is worth taking more space here to investigate further properties of these matrices. Recall that we defined the bias-count matrix $C(\mathbf{s}) = [c_{ij}]_{i,j\in\mathcal{N}}$, the probability-bias matrix $B_{\text{prob}}(\mathbf{s}) = [b_{ij}]_{i,j\in\mathcal{N}}$, and the skew-bias matrix $B_{\text{skew}}(\mathbf{s}) = [\beta_{ij}]_{i,j\in\mathcal{N}}$.

Lemma 3.1. *Let* $\mathbf{s}$ *and* $\mathbf{s}'$ *be sequences, and consider the following statements:*

1. $C(\mathbf{s}) = C(\mathbf{s}')$
2. $B_{prob}(\mathbf{s}) = B_{prob}(\mathbf{s}')$
3. $B_{skew}(\mathbf{s}) = B_{skew}(\mathbf{s}')$

Then statements (1) and (2) are equivalent. Further, (1) and (2) imply (3).

Proof. First, note that that (1) implies (2) since B_{prob} was defined solely in terms of C. Similarly, (2) implies (3) since B_{skew} was defined solely in terms of B_{prob}. All that remains to be shown is that (2) implies (1). To see this, note that the spike count $c_i(\mathbf{s})$ can be recovered from the diagonal entry $b_{ii}(\mathbf{s})$ of $B_{\text{prob}}(\mathbf{s})$:

$$b_{ii}(\mathbf{s}) = \frac{c_{ii}(\mathbf{s})}{c_i(\mathbf{s})c_i(\mathbf{s})} = \frac{\binom{c_i(\mathbf{s})}{2}}{c_i(\mathbf{s})c_i(\mathbf{s})} = \frac{1}{2}\left(1 - \frac{1}{c_i(\mathbf{s})}\right) \implies c_i(\mathbf{s}) = \frac{1}{1 - 2b_{ii}(\mathbf{s})}.$$

Knowing each spike count now allows us to recover any entry of $C(\mathbf{s})$:

$$b_{ij}(\mathbf{s}) = \frac{c_{ij}(\mathbf{s})}{c_i(\mathbf{s})c_j(\mathbf{s})} \implies c_{ij}(\mathbf{s}) = b_{ij}(\mathbf{s})c_i(\mathbf{s})c_j(\mathbf{s}).$$

Hence, $C(\mathbf{s}) = C(\mathbf{s}')$ if $B_{\text{prob}}(\mathbf{s}) = B_{\text{prob}}(\mathbf{s}')$. Therefore, (2) implies (1), as claimed. This completes the proof. □

Note that statement (3) from the above lemma is not equivalent to statements (1) and (2). This is easily seen, for example, with the pure sequences 12 and 1122:

$$B_{\text{skew}}(12) = B_{\text{skew}}(1122) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

but

$$C(12) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \neq \begin{bmatrix} 1 & 4 \\ 0 & 1 \end{bmatrix} = C(1122).$$

The reason for this is that $B_{\text{skew}}(\mathbf{s})$ does not contain sufficient information to recover the spike counts since $\beta_{ii}(\mathbf{s}) = 0$ for all neurons $i \in \mathcal{N}$ and all sequences $\mathbf{s}$.

3.1 Relationship to center-of-mass vectors

Bias matrices were introduced in an attempt to deal with issues associated with center-of-mass vectors, but these two concepts were reached through very different approaches. Still, a surprising relationship exists between them: The center-of-mass vector center($\mathbf{s}$) can be computed from the bias-count matrix $C(\mathbf{s})$.

Lemma 3.2. *If $\mathbf{s} \in \mathcal{S}$ is a sequence and $i \in \operatorname{supp}(\mathbf{s})$ is an active neuron, then* $\operatorname{center}_i(\mathbf{s}) = 1 + \frac{1}{c_i(\mathbf{s})} \sum_{j=1}^{n} c_{ji}(\mathbf{s})$.

Proof. Let $\ell = \operatorname{len}(\mathbf{s})$. Also, let $c_{ji}^{(k)}(\mathbf{s})$ denote the number of times neuron j spikes before the \$k\$-th spike in the sequence $\mathbf{s}$ provided that $s_k = i$:

$$c_{ji}^{(k)}(\mathbf{s}) := |\{k' < k \mid s_{k'} = j \text{ and } s_k = i\}|.$$

Observe that

$$\sum_{k=1}^{\ell} c_{ji}^{(k)}(\mathbf{s}) = c_{ji}(\mathbf{s}) \qquad \text{and} \qquad \sum_{j=1}^{n} c_{ji}^{(k)}(\mathbf{s}) = \begin{cases} k-1, & s_k = i; \\ 0, & s_k \neq i. \end{cases}$$

Further, since we know $C(\mathbf{s})$, we know $c_{ii}(\mathbf{s})$ for each $i \in \mathcal{N}$. Recalling that $c_{ii}(\mathbf{s}) = \binom{c_i(\mathbf{s})}{2}$, we note that $c_i(\mathbf{s}) = \frac{1}{2} + \frac{1}{2}\sqrt{1 + 8c_{ii}(\mathbf{s})}$. Moreover, knowing $c_{ii}(\mathbf{s})$ for each $i \in \mathcal{N}$ allows us to find ℓ as $\ell = \sum_{i \in \mathcal{N}} c_i(\mathbf{s})$. Putting these facts together, we see that

$$\begin{aligned} \sum_{j=1}^{n} c_{ji}(\mathbf{s}) &= \sum_{j=1}^{n} \sum_{k=1}^{\ell} c_{ji}^{(k)}(\mathbf{s}) \\ &= \sum_{k=1}^{\ell} \sum_{j=1}^{n} c_{ji}^{(k)}(\mathbf{s}) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{k \in L_i(\mathbf{s})} (k-1) \\
&= \sum L_i(\mathbf{s}) - c_i(\mathbf{s}) \\
&= c_i(\mathbf{s})[\text{center}_i(\mathbf{s}) - 1].
\end{aligned}
$$

Because $c_i(\mathbf{s}) \neq 0$ by assumption, we can solve for $\text{center}_i(\mathbf{s})$ to obtain the desired result. □

In particular, the above lemma says that $\text{center}(\mathbf{s}_1) = \text{center}(\mathbf{s}_2)$ if $C(\mathbf{s}_1) = C(\mathbf{s}_2)$. However, the converse to this statement is not true. That is, the center-of-mass vector $\text{center}(\mathbf{s})$ does not contain enough information to compute the bias-count matrix. For example, consider the sequences $\mathbf{s}_1 = 123123$ and $\mathbf{s}_2 = 211332$. Then

$$
\text{center}(\mathbf{s}_1) = \text{center}(\mathbf{s}_2) = \begin{bmatrix} 5/2 \\ 7/2 \\ 9/2 \end{bmatrix}
$$

but

$$
C(\mathbf{s}_1) = \begin{bmatrix} 1 & 3 & 3 \\ 1 & 1 & 3 \\ 1 & 1 & 1 \end{bmatrix} \neq \begin{bmatrix} 1 & 2 & 4 \\ 2 & 1 & 2 \\ 0 & 2 & 1 \end{bmatrix} = C(\mathbf{s}_2).
$$

As such, the bias-count matrix $C(\mathbf{s})$ is in fact a generalization of the center-of-mass vector $\text{center}(\mathbf{s})$.

One might also suspect that the center-of-mass vector $\gamma(\mathbf{s})$ can be recovered from the skew-bias matrix $B_{\text{skew}}(\mathbf{s})$. However, upon closer inspection of the previous lemma, we notice that the bias-count matrix was used to recover the spike-count vector, which cannot be done when starting with the skew-bias matrix. As the next proposition shows, however, we can reconstruct the center-of-mass vector $\gamma(\mathbf{s})$ from the skew-bias $B_{\text{skew}}(\mathbf{s})$ if we also know the sequence's firing-rate vector $\rho(\mathbf{s})$.

Proposition 3.3. *If* $\mathbf{s} \in \mathcal{S}$ *is a sequence, then*

$$\gamma(\mathbf{s}) = B_{skew}(\mathbf{s})^T \rho(\mathbf{s}).$$

Proof. Let $\ell = \text{len}(\mathbf{s})$, and assume that $i \in \text{supp}(\mathbf{s})$. Recall that $\gamma_i(\mathbf{s}) = \frac{2\,\text{center}_i(\mathbf{s})-1}{\ell} - 1$, from which we see that

$$\text{center}_i(\mathbf{s}) = \frac{\ell}{2}[\gamma_i(\mathbf{s}) + 1] + \frac{1}{2}.$$

We now want to rewrite the previous lemma to give us an expression for $\gamma_i(\mathbf{s})$ in terms of the skew-biases $\beta_{ij}(\mathbf{s})$. Dropping the $\mathbf{s}$ dependence from the notation, the above equation and the previous lemma yield the following:

$$\begin{aligned}
\frac{\ell}{2}(\gamma_i + 1) + \frac{1}{2} &= \text{center}_i \\
&= 1 + \frac{1}{c_i}\sum_{j=1}^{n} c_{ji} \\
&= 1 + \frac{1}{c_i}\Big(c_{ii} + \sum_{j\neq i} c_{ji}\Big) \\
&= 1 + \frac{1}{c_i}\Big(\frac{c_i(c_i-1)}{2} + \sum_{j\neq i} \frac{c_i c_j}{2}(\beta_{ji}+1)\Big) \\
&= 1 + \frac{1}{2}\Big(c_i - 1 + \sum_{j\neq i} c_j(\beta_{ji}+1)\Big) \\
&= 1 + \frac{1}{2}\Big(\big[c_i + \sum_{j\neq i} c_j\big] - 1 + \sum_{j\neq i} c_j\beta_{ji}\Big) \\
&= 1 + \frac{1}{2}\Big(\ell - 1 + \sum_{j=1}^{n} c_j\beta_{ji}\Big),
\end{aligned}$$

where we have recalled in the last line that $\beta_{ii} = 0$ and $\sum_{j=1}^{n} c_j = \ell$. Solving for γ_i

and including the $\mathbf{s}$ dependence into our notation, we obtain

$$\gamma_i(\mathbf{s}) = \sum_{j=1}^{n} \beta_{ji}(\mathbf{s})\rho_j(\mathbf{s})$$

since $\rho_i(\mathbf{s}) = \frac{c_i(\mathbf{s})}{\ell}$. Note that although we derived this equation using the assumption that $c_i(\mathbf{s}) > 0$, it still holds if $c_i(\mathbf{s}) = 0$ because in this case we have $\gamma_i(\mathbf{s}) = 0$ and $\beta_{ij}(\mathbf{s}) = 0$ for all j. Since $\sum_{j=1}^{n} \beta_{ji}(\mathbf{s})\rho_j(\mathbf{s})$ is the entry in row i of $B_{\text{skew}}(\mathbf{s})^T\rho(\mathbf{s})$, this completes the proof. □

If we substitute the firing-rate vector $\rho(\mathbf{s})$ in Proposition 3.3 for an arbitrary firing-rate vector, then the skew-bias matrix can be used to generate other center-of-mass vectors and orderings.

Definition 3.4. Given a sequence $\mathbf{s}$ and a firing-rate column vector $\rho = (\rho_1, \ldots, \rho_n)$, define the *center of mass of* $\mathbf{s}$ *induced by* ρ to be $\gamma_{\mathbf{s}}(\rho) := B_{\text{skew}}(\mathbf{s})^T\rho$. Further, define the corresponding *center-of-mass ordering of* $\mathbf{s}$ *induced by* ρ, denoted $\sigma_{\mathbf{s}}(\rho)$, just as with the center-of-mass ordering of a sequence: If $\gamma_{\mathbf{s}}(\rho) = (g_1, \ldots, g_n)$, then $\sigma_{\mathbf{s}}(\rho) = (i_1, \ldots, i_n)$ where $\{i_1, \ldots, i_n\} = [n]$ are such that $g_{i_1}(\mathbf{s}) \leq \cdots \leq g_{i_n}(\mathbf{s})$ and, in the case that $g_{i_k}(\mathbf{s}) = g_{i_{k+1}}(\mathbf{s})$, such that $i_k < i_{k+1}$.

Example 3.5. Recall that the sequence $\mathbf{s} = 12444433131124$ has skew-bias matrix

$$B_{\text{skew}}(\mathbf{s}) = \begin{bmatrix} 0 & 1/4 & -1/3 & -1/5 \\ -1/4 & 0 & 0 & 1/5 \\ 1/3 & 0 & 0 & -3/5 \\ 1/5 & 1/5 & 3/5 & 0 \end{bmatrix}.$$

The firing-rate vector of $\mathbf{s}$ is $\rho(\mathbf{s}) = \frac{1}{14}[4, 2, 3, 5]$ and $\sigma(\mathbf{s}) = (4, 2, 1, 3)$. With firing-rate

vectors $\rho_1 = \frac{1}{4}[1,1,1,1]$ and $\rho_2 = \frac{1}{3}[1,1,1,0]$, we find that

$$\gamma_{\mathbf{s}}(\rho_1) = \begin{bmatrix} 17/240 \\ 1/80 \\ 1/15 \\ -1/4 \end{bmatrix} \longrightarrow \sigma_{\mathbf{s}}(\rho_1) = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \end{bmatrix} \qquad \text{and} \qquad \gamma_{\mathbf{s}}(\rho_2) = \begin{bmatrix} 1/36 \\ 1/12 \\ -1/9 \\ -1/3 \end{bmatrix} \longrightarrow \sigma_{\mathbf{s}}(\rho_2) = \begin{bmatrix} 4 \\ 3 \\ 1 \\ 2 \end{bmatrix}.$$

So, ρ_1 and ρ_2 induce the orderings $(4,2,3,1)$ and $(4,3,1,2)$, respectively. In particular, notice that the activation of neuron 4 changed the induced ordering of $\{1,2,3\}$.

In the case of a pure sequence, this process can result in only one neuronal ordering, as the following lemma shows.

Lemma 3.6. *If* $\mathbf{s}$ *is a pure sequence, then* $\sigma_{\mathbf{s}}(\rho) = \sigma(\mathbf{s})$ *for each firing-rate vector* $\rho = (\rho_1, \ldots, \rho_n)$ *satisfying* $\rho_i > 0$ *for each* $i \in \mathcal{N}$.

Proof. Without loss of generality, we can assume that $\mathbf{s} = \sigma(\mathbf{s})$ since $B_{\text{skew}}(\mathbf{s}) = B_{\text{skew}}(\sigma(\mathbf{s}))$ whenever $\mathbf{s}$ is pure. For each neuron $i \in [n-1]$, we know that $\beta_{js_i}(\mathbf{s}) = \beta_{js_{i+1}}(\mathbf{s})$ for each $j \in \mathcal{N} \setminus \{s_i, s_{i+1}\}$ since $\mathbf{s}$ is pure and since s_i and s_{i+1} are adjacent spikes (not equal to j); additionally, we know that $\beta_{s_is_i}(\mathbf{s}) = \beta_{s_{i+1}s_{i+1}}(\mathbf{s}) = 0$ and that $\beta_{s_is_{i+1}}(\mathbf{s}) = 1$ and $\beta_{s_{i+1}s_i}(\mathbf{s}) = -1$. Letting $\gamma_{\mathbf{s}}(\rho) = (\gamma_1, \ldots, \gamma_n)$, these facts tell us that

$$\gamma_{s_i} = \sum_{j=1}^{n} \beta_{js_i}(\mathbf{s})\rho_j < \sum_{j=1}^{n} \beta_{js_{i+1}}(\mathbf{s})\rho_j = \gamma_{s_{i+1}}.$$

Hence, regardless of the choice of ρ, we always have that $\gamma_{s_1} < \cdots < \gamma_{s_n}$. In particular, letting $\rho = \rho(\mathbf{s})$, we find that $\sigma_{\mathbf{s}}(\rho(\mathbf{s})) = \sigma(\mathbf{s})$, thus completing the proof. □

3.2 Permutations of sequences

To understand a bias matrix A, it would be useful to know which sequences give rise to that bias matrix; that is, we would like to be able to describe the set $B_{\text{skew}}^{-1}(A) \subseteq \mathcal{S}$.

We can "fill in" part of this set if we know a sequence in that set, that is, a sequence $\mathbf{s}$ such that $B_{\text{skew}}(\mathbf{s}) = A$.

Consider the sequence $\mathbf{s} = 12321$ and its bias-count matrix

$$C(\mathbf{s}) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

If we flip any two adjacent spikes, the result in the bias-count matrix will be that exactly two entries are modified. For instance, if we flip the first two spikes, the bias-count matrix becomes

$$C(21321) = \begin{bmatrix} 1 & 1 & 1 \\ 3 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Since we want the bias-count matrix to remain unchanged if we are to find another sequence in $B_{\text{skew}}^{-1}(B_{\text{skew}}(\mathbf{s}))$, we can try to undo the effect of the first flip by flipping another pair of spikes; that is, since we made a subsequence of the form 12 into one of the form 21, we can look for a subsequence of the form 21 to change into 12. Here, the last two spikes are 21; changing these and the first two, we get the sequence $\mathbf{s}' = 21312$. Now, $C(\mathbf{s}') = C(\mathbf{s})$. So, indeed, it is sometimes possible to find another sequence with the same biases by simply permuting the spikes of the original sequence. It will be convenient to have some notation and terminology for such an operation:

Definition 3.7. For a sequence $\mathbf{s} = (s_1, \ldots, s_\ell)$ and a permutation $\pi \in \text{Perms}([\ell])$, define the *shuffling of* $\mathbf{s}$ *by* π to be $\mathbf{s}^\pi := (s_{\pi(1)}, \ldots, s_{\pi(\ell)})$.

Example 3.8. Let $\mathbf{s} = 1212333$. The following table shows permutations of $[\text{len}(\mathbf{s})] = [7]$ along with the corresponding shuffled sequence.

Permutation	Shuffled sequence
$\pi_1 = (1\ 7)(2\ 6)(3\ 5)$	$\mathbf{s}^{\pi_1} = 3332121$
$\pi_2 = (2\ 3)$	$\mathbf{s}^{\pi_2} = 1122333$
$\pi_3 = (1\ 2)(3\ 4)$	$\mathbf{s}^{\pi_3} = 2121333$
$\pi_4 = (1\ 3)(5\ 6\ 7)$	$\mathbf{s}^{\pi_4} = 1212333$

Note that the non-trivial permutation π_4 induces a shuffled sequence $\mathbf{s}^{\pi_4}$ that is equal to the original sequence $\mathbf{s}$.

Definition 3.9. For a sequence $\mathbf{s} = (s_1, \ldots, s_\ell)$ and a collection ω of neurons, define the *restriction of* $\mathbf{s}$ *to* ω to be $\mathbf{s}|_\omega := s_{I_\omega}$ where $I_\omega := \cup_{i\in\omega} L_i(\mathbf{s})$.

Example 3.10. Let $\mathbf{s} = 12444433131124$, and let $\omega_1 = \{1, 2\}$ and $\omega_2 = \{3, 4\}$. Then $\mathbf{s}_{\omega_1} = 121112$ and $\mathbf{s}_{\omega_2} = 44443334$.

Lemma 3.11. *Let* $\mathbf{s} = (s_1, \ldots, s_\ell)$ *be a sequence, and let* $\pi = (k_1\ k_1 + 1)(k_2\ k_2 + 1)$ *be a permutation such that* $s_{k_1} = s_{k_2+1}$ *and* $s_{k_2} = s_{k_1+1}$. *Then* $C(\mathbf{s}) = C(\mathbf{s}^\pi)$.

Proof. Let $\mathbf{s} = (s_1, \ldots, s_\ell)$ be a sequence, and let $\pi = (k_1\ k_1 + 1)(k_2\ k_2 + 1)$ be a permutation of $[\ell]$ such that $s_{k_1} = s_{k_2+1}$ and $s_{k_2} = s_{k_1+1}$. Let $i_1 = s_{k_1} \in \mathcal{N}$ and $i_2 = s_{k_2} \in \mathcal{N}$, and let $N = \mathcal{N} \setminus \{i_1, i_2\}$. Observe that $\mathbf{s}|_N = \mathbf{s}^\pi|_N$ since π fixes the indices of the spikes from neurons in N; in particular, this tells us that $c_{j_1j_2}(\mathbf{s}) = c_{j_1j_2}(\mathbf{s}|_N) = c_{j_1j_2}(\mathbf{s}^\pi|_N) = c_{j_1j_2}(\mathbf{s}^\pi)$.

We now want to show that $c_{ij}(\mathbf{s}) = c_{ij}(\mathbf{s}^\pi)$ and $c_{ji}(\mathbf{s}) = c_{ji}(\mathbf{s}^\pi)$ if $\{i, j\} \cap \{i_1, i_2\} \neq \emptyset$. Recalling that $c_{ij} = |L_{ij}|$, we proceed by listing the relationships between index pairs in the sets L_{ij} and L_{ji}. Suppose that $m \in [\ell] \setminus \{k_1, k_1 + 1, k_2, k_2 + 1\}$ is a spike index and that $j := s_m \in N$. Then the following relationships hold:

$$(k_1, m) \in L_{i_1j}(\mathbf{s}) \iff (k_1 + 1, m) \in L_{i_1j}(\mathbf{s}^\pi)$$

$$(k_2 + 1, m) \in L_{i_1j}(\mathbf{s}) \iff (k_2, m) \in L_{i_1j}(\mathbf{s}^\pi)$$

$$(k_1 + 1, m) \in L_{i_2j}(\mathbf{s}) \iff (k_1, m) \in L_{i_2j}(\mathbf{s}^\pi)$$

$$
\begin{aligned}
(k_2, m) \in L_{i_2 j}(\mathbf{s}) &\iff (k_2 + 1, m) \in L_{i_2 j}(\mathbf{s}^\pi) \\
(m, k_1) \in L_{j i_1}(\mathbf{s}) &\iff (m, k_1 + 1) \in L_{j i_1}(\mathbf{s}^\pi) \\
(m, k_2 + 1) \in L_{j i_1}(\mathbf{s}) &\iff (m, k_2) \in L_{j i_1}(\mathbf{s}^\pi) \\
(m, k_1 + 1) \in L_{j i_2}(\mathbf{s}) &\iff (m, k_1) \in L_{j i_2}(\mathbf{s}^\pi) \\
(m, k_2) \in L_{j i_2}(\mathbf{s}) &\iff (m, k_2 + 1) \in L_{j i_2}(\mathbf{s}^\pi) \\
(k_1, k_1 + 1), (k_1, k_2) \in L_{i_1 i_2}(\mathbf{s}) &\iff (k_1 + 1, k_2 + 1), (k_2, k_2 + 1) \in L_{i_1 i_2}(\mathbf{s}^\pi) \\
(k_1 + 1, k_2 + 1), (k_2, k_2 + 1) \in L_{i_2 i_1}(\mathbf{s}) &\iff (k_1, k_1 + 1), (k_1, k_2) \in L_{i_2 i_1}(\mathbf{s}^\pi)
\end{aligned}
$$

This completes the proof since each line above indicates that the corresponding index-pair sets are of the same size for both $\mathbf{s}$ and $\mathbf{s}^\pi$. □

Since a sequence of such permuations can be applied recursively, one might wonder whether all sequences with equal bias-count matrices can be obtained from a single sequence by repeated application of such permutations. This is not the case: For example, if $\mathbf{s} = 123231$ and $\mathbf{s}' = 231123$, then $C(\mathbf{s}) = C(\mathbf{s}')$ but no such permutation $\pi = (i\ i+1)(j\ j+1)$ exists such that $\mathbf{s}' = \mathbf{s}^\pi$ since there is no pair $\{i, j\}$ with $i \neq j$ such that $s_i = s_{j+1}$ and $s_{i+1} = s_j$. More succinctly, if $B_{\text{prob}}(\mathbf{s}) = B_{\text{prob}}(\mathbf{s}')$, it is not necessarily the case that $\mathbf{s}' = \mathbf{s}^\pi$ for some permutation π.

We can also describe a broader class of permutations that fixes the center-of-mass vectors of a given sequence. Generally speaking, elements of this class will not fix the corresponding bias matrices.

Lemma 3.12. *If* $\mathbf{s} = (s_1, \ldots, s_\ell)$ *is a sequence and* $\pi = (k_1\ k_1 + 1) \cdots (k_m\ k_m + 1) \in \text{Perms}([l])$ *with* $k_{a+1} > k_a + 1$ *for all* $a \in \{1, \ldots, m-1\}$*, then* $\text{center}(\mathbf{s}) = \text{center}(\mathbf{s}^\pi)$ *if* $\{s_{k_1}, \ldots, s_{k_m}\} = \{s_{k_1+1}, \ldots, s_{k_m+1}\}$ *and* $|\{s_{k_1}, \ldots, s_{k_m}\}| = m$.

Proof. Suppose that $i \in \{s_1, \ldots, s_k\}$. Then $i = s_{k_b} = s_{k_d+1}$ for some $b, d \in [m]$.

Noting that $L_i(\mathbf{s}^\pi) = [L_i(\mathbf{s}) \setminus \{k_b, k_d + 1\}] \cup \{k_b + 1, k_d\}$, we see that

$$\begin{aligned}\text{center}_i(\mathbf{s}) &= \text{center}_i(\mathbf{s}) + \frac{[k_b + (k_d + 1)] - [k_b + (k_d + 1)]}{c_i(\mathbf{s})} \\ &= \text{center}_i(\mathbf{s}) + \frac{[(k_b + 1) + k_d] - [k_b + (k_d + 1)]}{c_i(\mathbf{s}^\pi)} \\ &= \text{center}(\mathbf{s}^\pi).\end{aligned}$$

If $i \notin \{s_1, \ldots, s_k\}$, then $\text{center}_i(\mathbf{s}) = \text{center}_i(\mathbf{s}^\pi)$ since $L_i(\mathbf{s}^\pi) = L_i(\mathbf{s})$. Hence, $\text{center}_i(\mathbf{s}) = \text{center}_i(\mathbf{s}^\pi)$ for all $i \in \mathcal{N}$, thus completing the proof. □

Example 3.13. Let $\mathbf{s} = 123123$ and $\pi = (1\ 2)(3\ 4)(5\ 6)$. Then $\mathbf{s}$ and π meet the stipulations of the previous lemma since $\{s_1, s_3, s_5\} = \{s_2, s_4, s_6\} = \{1, 2, 3\}$. Noting that $\mathbf{s}^\pi = 211332$, we see that

$$\text{center}(\mathbf{s}) = \text{center}(\mathbf{s}^\pi) = \begin{bmatrix} 7/2 \\ 5/2 \\ 9/2 \end{bmatrix}.$$

3.3 The fundamental polytope

The bias matrix arose in an attempt to capture the tendency of pairs of neurons to fire in a particular order. These tendencies can be computed from any individual sequence, but the pairwise nature of the biases hints at an underlying network of neurons with pairwise preferences for spiking in particular orders. From this perspective, a sequence is really just a (noisy) manifestation of the neural network's biases. As such, it may be useful to understand the properties that characterize all bias matrices. By considering the biases that arise from sequences, we aim to derive a bias-matrix formulation that is independent of any individual sequence. In this section, we develop such a formulation and conjecture that each collection of biases

in this formulation is realizable as the collection of biases of an actual sequence.

We begin by consideration of the bias-count matrix $C(\mathbf{s}) = [c_{ij}(\mathbf{s})]_{i,j\in\mathcal{N}}$. We already know a few properties that must be satisfied:

- Its entries are natural numbers: $c_{ij} \in \mathbb{N}_0$ for all $i, j \in \mathcal{N}$.
- Its diagonal entries are binomial coefficients: $c_{ii} = \binom{c_i}{2}$ for all $i \in \mathcal{N}$.
- For all neurons $i, j \in \mathcal{N}$ with $i \neq j$, $c_{ij} + c_{ji} = c_i c_j$.

But we moved from the bias-count matrix to the skew-bias matrix because we want the individual bias entries to have meaning independent of other entries. When the above properties are translated into the skew-bias matrix, we find only two properties:

- It is skew-symmetric: $\beta_{ij} = -\beta_{ji}$ for all $i, j \in \mathcal{N}$.
- Its entries are bounded and rational: $\beta_{ij} \in [-1, 1] \cap \mathbb{Q}$ for all $i, j \in \mathcal{N}$.

Though the properties listed here do not fully characterize the skew-biases in general (as we will see below), they do provide a convenient starting place from which we can further restrict as we look for additional properties. The skew nature of the matrices $B_{\text{skew}}(\mathbf{s})$ allows us to describe the matrix using only the above-diagonal entries of each matrix.

Definition 3.14. The *bias space* $\mathfrak{B}(\mathcal{N})$ of the neuron set $\mathcal{N} = [n]$ is the $\binom{n}{2}$-dimensional hypercube $[-1, 1]^{\binom{n}{2}}$ with coordinates indexed by neuron pairs $(i, j) \in \mathcal{N}^2$ with $i < j$. When $\mathbf{s}$ is a sequence, the vector $[\beta_{ij}(\mathbf{s})]_{i<j}$ consisting of the upper-triangular portion of $B_{\text{skew}}(\mathbf{s})$ is the corresponding point in the bias space. We will abuse terminology and refer to $B_{\text{skew}}(\mathbf{s})$ as being in the bias space for notational/terminological convenience.

Definition 3.15. A point $x \in \mathfrak{B}(\mathcal{N})$ is *obtainable* if x is in the closure of the set $\{B_{\text{skew}}(\mathbf{s}) \mid \mathbf{s} \in \mathcal{S}\}$.

Which points in $\mathfrak{B}(\mathcal{N})$ are obtainable? Let us first consider one of the simplest classes of sequences: those with exactly two active neurons. In this case—that is, when $\mathcal{N} = [2]$—the only bias that we consider is β_{12}; further, the bias space is just the interval $[-1, 1]$. Here, we find that the biases are actually dense in the bias space.

Lemma 3.16. *For any $x \in [-1, 1]$ and any $\epsilon > 0$, there exists a sequence $\mathbf{s} \in \mathcal{S}$ such that* $\text{supp}(\mathbf{s}) = [2]$ *and* $|\beta_{12}(\mathbf{s}) - x| < \epsilon$.

Proof. Let $x \in [-1, 1]$ and $\epsilon > 0$ be given. Choose $\ell \in \mathbb{N}$ such that $\frac{2}{\ell-1} < \epsilon$, and pick $k \in \mathbb{N}$ as $k = \left\lfloor \frac{1}{2}(1+x)(\ell-1)+1 \right\rfloor$. Note that $0 \leq 1 + x \leq 2$ since $-1 \leq x \leq 1$; so, $0 \leq \frac{1}{2}(1+x)(\ell-1) \leq \ell - 1$ and, thus, $k \in [\ell]$. Moreover, $\left|k - \frac{1}{2}(1-x)\ell\right| \leq 1$ by choice of k. Define the sequence $\mathbf{s} = (s_1, \ldots, s_\ell)$ by $s_k = 2$ and $s_m = 1$ for $m \in [\ell] \setminus \{k\}$. Now, $c_{12}(\mathbf{s}) = k - 1$, which implies that $b_{12}(\mathbf{s}) = \frac{c_{12}(\mathbf{s})}{c_1(\mathbf{s})c_2(\mathbf{s})} = \frac{k-1}{(\ell-1)\cdot 1} = \frac{k-1}{\ell-1}$ and that $\beta_{12}(\mathbf{s}) = 2b_{12}(\mathbf{s}) - 1 = \frac{2k-2}{\ell-1} - 1$. Now,

$$\begin{aligned}
|\beta_{12}(\mathbf{s}) - x| &= \left| \frac{2k-2}{\ell-1} - 1 - x \right| \\
&= \left| \frac{2k - 2 - (1+x)(\ell-1)}{\ell-1} \right| \\
&= \frac{2}{\ell-1} \left| k - \left(\tfrac{1}{2}(1+x)(\ell-1) + 1 \right) \right| \\
&\leq \frac{2}{\ell-1} \\
&< \epsilon,
\end{aligned}$$

thus completing the proof. □

But it is not generally the case that every point in the bias space is obtainable.

To get some intuition about why some points are not obtainable, consider the matrix

$$A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}.$$

Suppose that $A = B_{\text{skew}}(\mathbf{s})$ for some sequence $\mathbf{s}$. Because all off-diagonal entries are in $\{1, -1\}$, we must have that $\mathbf{s}$ is a pure sequence. Considering the entries more closely, we also find the following:

- $(a_{12} = 1)$ All spikes from neuron 1 precede all spikes from neuron 2.
- $(a_{23} = 1)$ All spikes from neuron 2 precede all spikes from neuron 3.
- $(a_{13} = -1)$ All spikes from neuron 3 precede all spikes from neuron 1.

The first two points ($a_{12} = a_{23} = 1$) tell us that all spikes from neuron 1 must precede all spikes from neuron 3. But this is a contradiction to the third observation! Hence, we cannot have that $A = B_{\text{skew}}(\mathbf{s})$ for any sequence $\mathbf{s}$. Indeed, we will see in Corollary 3.19 that A is not obtainable.

Although we have not yet shown that A is not obtainable, the point is well-taken: There are non-trivial relationships among the entries β_{ij}, β_{jk}, and β_{ik} (that is, among the collection of biases between the neurons i, j, and k). In fact, the above-observed property holds in a more general case: If $\mathbf{s} \in \mathcal{S}$ is pure, then we must have that $\beta_{ik} = 1$ if $\beta_{ij} = \beta_{jk} = 1$ for distinct neurons $i, j, k \in \mathcal{N}$. The logic for seeing that this holds is exactly the same as in the motivating example. As seen in the proof of Lemma 3.16, only one spike from the second neuron was necessary to obtain (in the sense of density) all possible biases between exactly two neurons. As such, we begin our investigation of biases among three neurons by considering sequences with exactly one spike from a third neuron.

Lemma 3.17. *For any sequence* $\mathbf{s} \in \mathcal{S}$ *with distinct neurons* $i, j, k \in \text{supp}(\mathbf{s})$ *and with* $c_k(\mathbf{s}) = 1$, $b_{ik}(1 - b_{jk}) \leq b_{ij} \leq 1 - b_{jk}(1 - b_{ik})$.

Proof. Since neuron k spikes only once, it is easy to construct a sequence that minimizes m_{ij} for fixed (rational) values of b_{ki} and b_{kj}. Since $b_{ki} = \frac{c_{ki}}{c_k c_i} = \frac{c_{ki}}{c_i}$ and $b_{kj} = \frac{c_{kj}}{c_k c_j} = \frac{c_{kj}}{c_j}$, we see that $c_{ki} = b_{ki}c_i$ and $c_{kj} = b_{kj}c_j$ are proportions of the total number of times that neurons i and j spike, respectively. In particular, when considering the restricted sequences containing k and only one other neuron, we find that there is only one possibility for each restricted sequence:

$$\mathbf{s}|_{\{i,k\}} = i^{c_{ik}} \cdot k \cdot i^{c_{ki}} \qquad \text{and} \qquad \mathbf{s}|_{\{j,k\}} = j^{c_{jk}} \cdot k \cdot j^{c_{kj}},$$

where we recall that the symbol $\cdot$ represents concatenation. In joining these restricted sequences to form the full sequence $\mathbf{s}$, we minimize b_{ij} by placing j before i whenever possible, resulting in the following:

$$\mathbf{s} = j^{c_{jk}} \cdot i^{c_{ik}} \cdot k \cdot j^{c_{kj}} \cdot i_{c_{ki}}.$$

It is now straightforward to count c_{ij}:

$$c_{ij} = c_{ik}c_{kj} = (b_{ik}c_i)(b_{kj}c_j) = b_{ik}(1 - b_{jk})c_i c_j.$$

Because $\mathbf{s}$ was chosen to minimize c_{ij}, dividing by the product $c_i c_j$ yields one of the desired inequalities: $b_{ij} \leq b_{ik}(1 - b_{jk})$. Similar reasoning allows us to maximize c_{ij} with $\mathbf{s}'$:

$$\mathbf{s}' = i^{c_{ik}} \cdot j^{c_{jk}} \cdot k \cdot i^{c_{ki}} \cdot j_{c_{kj}},$$

which results in the count

$$\begin{aligned} c_{ij} &= c_{ik}c_j + c_{ki}c_{kj} \\ &= b_{ik}c_ic_j + (b_{ki}c_i)(b_{kj}c_j) \\ &= (b_{ik} + b_{ki}b_{kj})c_ic_j \\ &= [b_{ik} + (1 - b_{ik})(1 - b_{jk})]c_ic_j \\ &= (1 - b_{jk} + b_{ik}b_{jk})c_ic_j \\ &= [1 - b_{jk}(1 - b_{ik})]c_ic_j, \end{aligned}$$

thus yielding the other desired inequality: $b_{ij} \geq 1 - b_{jk}(1 - b_{ik})$. This completes the proof. □

As it turns out, however, not every point in the bias space $\mathfrak{B}([3])$ is obtainable using only a single spike from the third neuron. Consider, for example, the sequence $\mathbf{s} = 3123$ and its bias matrix

$$B_{\text{skew}}(\mathbf{s}) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Suppose that there is some sequence $\mathbf{s}'$ such that $c_3(\mathbf{s}') = 1$ and $B_{\text{skew}}(\mathbf{s}') = B_{\text{skew}}(\mathbf{s})$. Since $\beta_{12}(\mathbf{s}') = 1$, the restricted sequence $\mathbf{s}'|_{[2]}$ must be pure. But then the single spike from neuron 3 cannot fall in the middle of the spikes for neuron 1 and in the middle of the spikes for neuron 2. Hence, no such $\mathbf{s}'$ can exist. Still, not all points in the bias space are obtainable even when we consider arbitrary sequences on three neurons. But this should not come as too much of a surprise: Our intuition for sequences says that if i follows j and k follows j, then we would expect for k to follow i. The inequalities in the lemma below are restrictions of such a form.

Proposition 3.18. *For any sequence* $\mathbf{s}$ *with neurons* $i, j, k \in \text{supp}(\mathbf{s})$ *such that* $i < j < k$*, the inequalities* $0 \leq b_{ij} + b_{jk} - b_{ik} \leq 1$ *hold.*

Proof. Let $\mathbf{s} = (s_1, \ldots, s_\ell)$ be a sequence. First, note that the following equality holds because every triple $\ell_i, \ell_j, \ell_k \in [\ell]$ with $s_{\ell_i} = i$, $s_{\ell_j} = j$, and $s_{\ell_k} = k$ (of which there are $c_i c_j c_k$) will produce exactly one of the six orderings ijk, jki, kij, ikj, jik, and kji:

$$c_{ijk} + c_{ikj} + c_{jik} + c_{jki} + c_{kij} + c_{kji} = c_i c_j c_k$$

We now show that the following relationships exist between pairwise-bias counts and third-order bias counts:

$$c_k c_{ij} = c_{ijk} + c_{ikj} + c_{kij}$$

$$c_i c_{jk} = c_{jki} + c_{jik} + c_{ijk}$$

$$c_j c_{ki} = c_{kij} + c_{kji} + c_{jki}$$

To see that these relationships hold, consider the first one. For each subsequence of $\mathbf{s}$ that is equal to (i, j)—of which there are c_{ij}—each of the c_k occurrences of k in $\mathbf{s}$ can produce only one of the triples (i, j, k), (i, k, j), and (k, i, j); moreover, each such combination must produce one such triple. Hence, the equality holds. Similar arguments work for each of the other two equations.

Summing the above equations, we get the following inequality:

$$\begin{aligned} c_k c_{ij} + c_i c_{jk} + c_j c_{ki} &= 2(c_{ijk} + c_{jki} + c_{kij}) + (c_{ikj} + c_{jik} + c_{kji}) \\ &= 2(c_{ijk} + c_{jki} + c_{kij} + c_{ikj} + c_{jik} + c_{kji}) \\ &\quad - (c_{ikj} + c_{jik} + c_{kji}) \\ &= 2c_i c_j c_k - (c_{ikj} + c_{jik} + c_{kji}) \end{aligned}$$

$$\leq 2c_ic_jc_k.$$

Dividing the resultant inequality by $c_ic_jc_k$ yields the inequality $b_{ij}+b_{jk}+b_{ki}\leq 2$. By considering the sum of the same equations again, we find another inequality:

$$\begin{aligned} c_kc_{ij}+c_ic_{jk}+c_jc_{ki} &= 2(c_{ijk}+c_{jki}+c_{kij})+(c_{ikj}+c_{jik}+c_{kji}) \\ &= (c_{ijk}+c_{jki}+c_{kij}+c_{ikj}+c_{jik}+c_{kji}) \\ &\quad +(c_{ijk}+c_{jki}+c_{kij}) \\ &= c_ic_jc_k+(c_{ikj}+c_{jik}+c_{kji}) \\ &\geq c_ic_jc_k. \end{aligned}$$

Dividing this by $c_ic_jc_k$ yields the inequality $b_{ij}+b_{jk}+b_{ki}\geq 1$. Putting these two inequalities together, we have that $1\leq b_{ij}+b_{jk}+b_{ki}\leq 2$. Recalling that $b_{ki}=1-b_{ik}$, this becomes $0\leq b_{ij}+b_{jk}-b_{ik}\leq 1$, thus completing the proof. □

Corollary 3.19. *For any sequence* $\mathbf{s}$ *with neurons* $i,j,k\in\operatorname{supp}(\mathbf{s})$ *such that* $i<j<k$*, the inequalities* $-1\leq\beta_{ji}+\beta_{kj}-\beta_{ik}\leq 1$ *hold.*

As sequences become less pure, our intuition about one neuron "following" another breaks down, and we run into tendencies that would be contradictory in the case of a pure sequence.

Example 3.20. Consider the sequence $\mathbf{s}=2331112123$. The skew-bias matrix of $\mathbf{s}$ is

$$B_{\text{skew}}(\mathbf{s})=\begin{bmatrix} 0 & 1/6 & -1/3 \\ -1/6 & 0 & 1/9 \\ 1/3 & -1/9 & 0 \end{bmatrix}.$$

Upon inspection, we see that (1) neuron 1 tends to precede neuron 2, (2) neuron 2 tends to precede neuron 3, and (3) neuron 3 tends to precede neuron 1.

Inequalities of the form in Corollary 3.19 essentially ensure that the pairwise-biases among any three neurons are consistent with each other (i.e., if j follows i and k follows j, then k follows i), a sort of pseudo-transitivity that is guaranteed when biases arise from an actual sequence. Further, the collection of all such inequalities carves out a polytope in the bias space, which we will call the *consistent-bias polytope*. Because three distinct neurons were required for this corollary, there are $\binom{|\mathcal{N}|}{3}$ such inequalities, each of which is really two distinct inequalities. Since the bias space itself is defined by inequalities of the form $-1 \leq \beta_{ij} \leq 1$ with $i < j$, this polytope can be described by a total of $2\binom{|\mathcal{N}|}{3} + 2\binom{|\mathcal{N}|}{2}$ inequalities:

- $\beta_{ij} + \beta_{jk} - \beta_{ik} \leq 1$ for all $i < j < k$,
- $-\beta_{ij} - \beta_{jk} + \beta_{ik} \leq 1$ for all $i < j < k$,
- $\beta_{ij} \leq 1$ for all $i < j$,
- $-\beta_{ij} \leq 1$ for all $i < j$.

The inequalities from Corollary 3.19 can be rearranged to bound the bias of i to fire before k if the other biases are known: $\beta_{ij} + \beta_{jk} - 1 \leq \beta_{ik} \leq \beta_{ij} + \beta_{jk} + 1$. This inequality is the direct equivalent to the intuition that k should follow i if k follows j and j follows i. But longer chains of this form do not create inequalities that are more restrictive than those arising from such chains of length three. For considering possible biases among more than three neurons, we start by looking at the simplest of such biases: those arising from pure sequences.

Definition 3.21. A *pure-bias matrix* is a matrix A such that $A = B_{\text{skew}}(\mathbf{s})$ for some pure sequence $\mathbf{s} \in \mathcal{S}$.

For each pure-bias matrix A, there is a sequence $\mathbf{s}$ with one spike per neuron such that $B_{\text{skew}}(\mathbf{s}) = A$. Let us start by considering the sequence $\mathbf{s} = (1, 2, \ldots, n)$. Here,

we see that i precedes j whenever $i < j$; as such, the corresponding entries of $B_{\text{skew}}(\mathbf{s})$ are $\beta_{ij}(\mathbf{s}) = 1$ and $\beta_{ji}(\mathbf{s}) = -1$. For example,

$$B_{\text{skew}}(1234) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & -1 & 0 \end{bmatrix}.$$

Indeed, up to row and column permutations, the bias matrix of each pure sequence is of this form, as the following lemma shows.

Lemma 3.22. *Suppose that $\mathbf{s} \in \mathcal{S}$ is pure. Then $B_{skew}(\mathbf{s}) = P^{-1}AP$ for $A = B_{skew}(12\cdots n)$ and some permutation matrix P.*

Proof. Let $\mathbf{s} \in \mathcal{S}$ be pure. Without loss of generality, we can assume that $\mathbf{s} = (s_1, \ldots, s_n)$ has one spike per neuron since, for a pure sequence, it is clear that $B_{\text{skew}}(\mathbf{s}) = B_{\text{skew}}(\sigma(\mathbf{s}))$. Note that $\beta_{s_i,s_j}(\mathbf{s}) = \beta_{ij}(12\cdots n)$ since s_i precedes s_j in $\mathbf{s}$ if and only if i precedes j in $12\cdots n$. Define $\pi \in \text{Perms}(\mathcal{N})$ by $\pi(s_i) = i$, and let $P = [p_{ij}]_{i,j\in\mathcal{N}}$ be the permutation matrix with entries

$$p_{ij} = \begin{cases} 1, & \pi(j) = i; \\ 0, & \text{otherwise.} \end{cases}$$

Letting $C = P^{-1}AP = [c_{ij}]_{i,j\in\mathcal{N}}$ and $D = AP = [d_{ij}]_{i,j\in\mathcal{N}}$, we find that

$$\begin{aligned} c_{s_i,s_j} &= \sum_{k=1}^{n} p_{k,s_i} d_{k,s_j} \\ &= \sum_{k=1}^{n} p_{k,s_i} \left(\sum_{\ell=1}^{n} a_{k,\ell} p_{\ell,s_j} \right) \\ &= \sum_{k=1}^{n} p_{k,s_i} a_{k,\pi(s_j)} \end{aligned}$$

$$= a_{\pi(s_i),\pi(s_j)}$$
$$= a_{ij}$$
$$= \beta_{ij}(12\cdots n)$$
$$= \beta_{s_i,s_j}(\mathbf{s}).$$

Therefore, $B_{\text{skew}}(\mathbf{s}) = P^{-1}AP$ as claimed, completing the proof. □

Pure sequences are extreme from the perspective of biases. Indeed, each bias corresponding to a pure sequence is a vertex of the the bias space hypercube. As we have already seen in this section, not all vertices of the bias space are legitimate biases. We now know exactly which vertices correspond to legitimate biases. Moreover, we have a simple count for how many vertices are obtainable biases: There is one for each permutation of the neuron set $\mathcal{N} = [n]$; that is, $n!$ of the $2^{\binom{n}{2}}$ bias-space vertices are obtainable biases.

But we can glean more from this investigation of pure biases. Consider two pure sequences with one spike per neuron that are as close to each other as possible, that is, sequences that differ by a single transposition of spikes. For the sake of example, we will consider $\mathbf{s} = 1234\cdots n$ and $\mathbf{s}' = \mathbf{s}^{(1\ 2)} = 2134\cdots n$. From the perspective of the neurons in $\mathcal{N} \setminus [2]$, neurons 1 and 2 both behave exactly the same (since both 1 and 2 precede all other spikes in both $\mathbf{s}$ and $\mathbf{s}'$). Many other sequences share this property with $\mathbf{s}$ and $\mathbf{s}'$; in fact, we can completely characterize the collection of all such sequences as $\{(s_1, \ldots, s_\ell, 3, 4, \ldots, n) \mid (s_1, \ldots, s_\ell) \in \mathcal{S}([2])\}$. That is, we can prepend an arbitrary sequence from the neuron set [2] to the beginning of the sequence $34\cdots n$. But we know something about the biases arising from the sequences in $\mathcal{S}([2])$: Every bias is obtainable! This tells us that the entire line joining $\mathbf{s}$ to $\mathbf{s}'$ in the bias space is obtainable as a bias. This is similarly true for any such pair of "adjacent" sequences

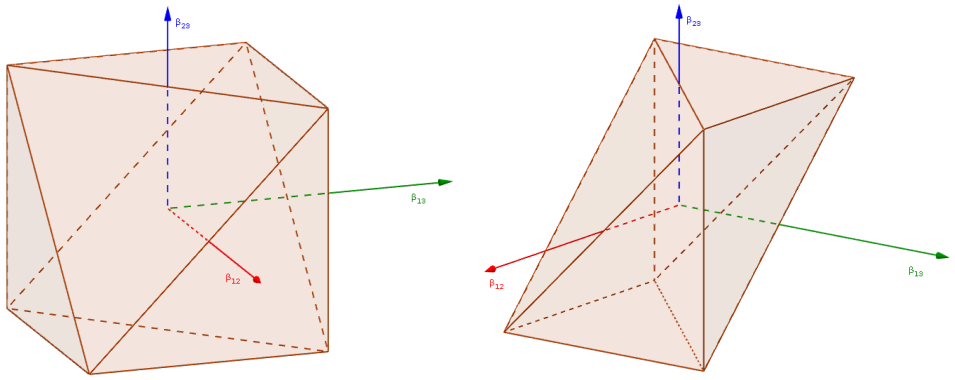


Figure 3.1: When $\mathcal{N} = [3]$, the fundamental polytope of $\mathcal{S}$ is 3-dimensional. Above, this polytope is shown below from two angles. In this case of only three neurons, the fundamental polytope is equal to the consistent-bias polytope. Note that $n = 3$ is a special case: **These polytopes are not equal in general.** Since they are equal here, we can easily write down the hyperplanes that serve as its boundaries: $-1 \leq \beta_{12} + \beta_{23} - \beta_{13} \leq 1$, $-1 \leq \beta_{12} \leq 1$, $-1 \leq \beta_{13} \leq 1$, and $-1 \leq \beta_{23} \leq 1$

with one spike per neuron. More surprisingly, each obtainable bias is in the convex hull of the collection of pure biases. Before showing this claim to be true, we name this polytope and show its relationship to the consistent-bias polytope.

Definition 3.23. The *fundamental polytope* of $\mathcal{S}$ is the convex hull of the set of pure biases arising from sequences in $\mathcal{S}$.

Lemma 3.24. *The consistent-bias polytope contains the fundamental polytope.*

Proof. Since the vertices of the fundamental polytope are obtainable biases, they must satisfy the constraints that specify the consistent-bias polytope; moreoever, since both polytopes are convex, we must have that the fundamental polytope is contained in the consistent-bias polytope. □

Theorem 3.25. *If* $\mathbf{s} \in \mathcal{S}$ *with* $\text{supp}(\mathbf{s}) = \mathcal{N}$*, then the bias corresponding to* $\mathbf{s}$ *is contained in the fundamental polytope.*

Proof. Let $\mathbf{t} = 12\cdots n$, and let $P_n = \{B_{\text{skew}}(\mathbf{t}^\pi) \mid \pi \in S_n\}$ be the collection of pure-bias matrices. Also, let $\mathbf{s} = (s_1, \ldots, s_\ell)$ be any sequence with $\text{supp}(\mathbf{s}) = \mathcal{N}$. We need only to show that $B_{\text{skew}}(\mathbf{s})$ is a convex combination of the elements of P_n. For each subsequence $\mathbf{s}'$ of $\mathbf{s}$ that is equal to $\mathbf{t}^\pi$ for some $\pi \in \text{Perms}([n])$, we have that $B_{\text{skew}}(\mathbf{s}') \in P_n$ by definition of P_n. Let $\text{Sub}(\mathbf{s}) = \{I \subseteq [\ell] \mid \mathbf{s}_I = \mathbf{t}^\pi \text{ for some } \pi \in \text{Perms}([n])\}$, and define the sum $D = [d_{ij}]$ to be

$$D = \frac{1}{\prod_{i=1}^n c_i(\mathbf{s})} \sum_{I \in \text{Sub}(\mathbf{s})} B_{\text{skew}}(\mathbf{s}_I).$$

Because $B_{\text{skew}}(\mathbf{s}_I) \in P_n$, the proof will be finished if $D = B_{\text{skew}}(\mathbf{s})$ and if the coefficients of its summands sum to 1.

To see that the latter of these requirements holds, first note that all coefficients are equal. Moreover, there are exactly $\prod_{i=1}^n c_i(\mathbf{s})$ summands in the definition of D: Choosing a subsequence of $\mathbf{s}$ that is and ordering of $\mathcal{N}$ is the same as choosing one index from each of the sets $L_i(\mathbf{s})$ for $i \in \mathcal{N}$, which can obviously be done in $\prod_{i=1}^n c_i(\mathbf{s})$ ways since $c_i(\mathbf{s}) = |L_i(\mathbf{s})|$ for each $i \in \mathcal{N}$.

To see that $D = B_{\text{skew}}(\mathbf{s})$, consider an arbitrary entry d_{ij}:

$$\begin{aligned}
d_{ij} &= \frac{1}{\prod_{k=1}^n c_k(\mathbf{s})} \sum_{I \in \text{Sub}(\mathbf{s})} \beta_{ij}(\mathbf{s}_I) \\
&= \frac{1}{\prod_{k=1}^n c_k(\mathbf{s})} \left(\sum_{\substack{I \in \text{Sub}(\mathbf{s}) \\ c_{ij}(\mathbf{s}_I)=1}} 1 + \sum_{\substack{I \in \text{Sub}(\mathbf{s}) \\ c_{ji}(\mathbf{s}_I)=1}} -1 \right) \\
&= \frac{1}{\prod_{k=1}^n c_k(\mathbf{s})} \left([c_{ij}(\mathbf{s}) - c_{ji}(\mathbf{s})] \cdot \prod_{k \notin \{i,j\}} c_k(\mathbf{s}) \right) \\
&= \frac{c_{ij}(\mathbf{s}) - c_{ji}(\mathbf{s})}{c_i(\mathbf{s}) c_j(\mathbf{s})} \\
&= \beta_{ij}(\mathbf{s})
\end{aligned}$$

Hence, we see that the fundamental polytope contains all biases. □

Though all biases are contained in the fundamental polytope, the question of whether or not the biases are dense in the fundamental polytope is still an open question.

Question 3.26. Is every point in the fundamental polytope obtainable?

3.4 The bias network of a sequence

Neuronal templates are appealing because they provide a simple combinatorial picture of neuronal sequences, but this representation is lacking in many aspects. The bias matrix can be used to define a more-robust combinatorial representation of these sequences.

Definition 3.27. Let $A = [a_{ij}]_{i,j\in\mathcal{N}}$ be a bias matrix. We define the *bias network of* A to be the simple digraph $G(A)$ with vertext set $\mathcal{N}$ and edge set $E(A) := \{(i,j) \mid a_{ij} > 0\}$. The *bias network of a sequence* $\mathbf{s}$ is $G(\mathbf{s}) := G(B_{\text{skew}}(\mathbf{s}))$ with edge set $E(\mathbf{s}) = E(B_{\text{skew}}(\mathbf{s}))$.

Definition 3.28. A *tournament* is a complete oriented graph.

Definition 3.29. A digraph D with edge-set E is *transitive* if $(v_1, v_3) \in E$ whenever $(v_1, v_2), (v_2, v_3) \in E$.

The vertices of a transitive tournament can be totally ordered by reachability (i.e., vertex v precedes vertex u in the ordering if (v, u) is an edge). The following lemma shows one way in which bias networks generalize neuronal templates: by unambiguously inducing a total ordering in the case of a pure sequence.

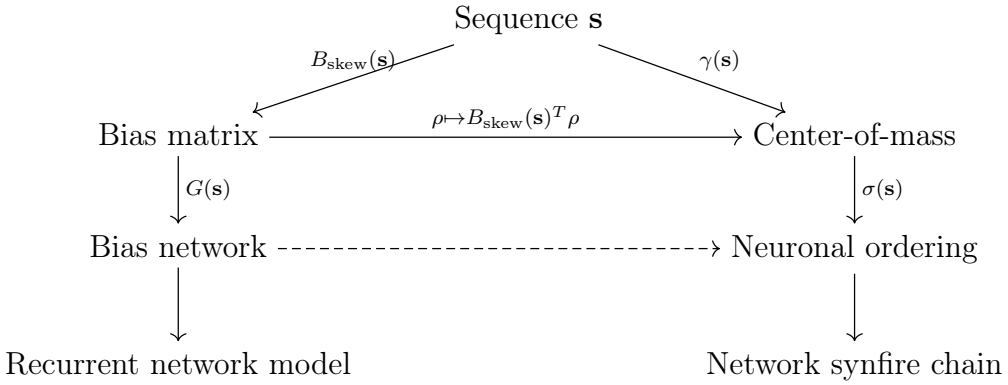


Figure 3.2: The bias network model. This diagram represents how the bias network serves as a generalization of neuronal templates. Starting with a sequence $\mathbf{s}$, the center-of-mass vector and ordering can be computed directly (right side). Alternately, the same vectors can be computed via the bias matrix; this also allows for the generation of other orderings by varying the firing-rate vector ρ used to generate the center-of-mass vector. The dashed line across the middle indicates that, in the case of pure sequences, the center-of-mass ordering is recoverable from the bias network, as shown in Lemma 3.30. At the bottom level, these two combinatorial objects can be used to generate sequences using recurrent networks and network synfire chains, topics beyond the scope of this paper.

Lemma 3.30. *If A is a pure-bias matrix, then $G(A)$ is a transitive tournament.*

Proof. Let $\mathcal{N} = [n]$. If $n = 1$, then the claim is trivially true. Let $\mathbf{s}$ be a sequence with $B_{\text{skew}}(\mathbf{s}) = A$, and suppose that $n > 1$. Since $\mathbf{s}$ is pure, some neuron $i \in \mathcal{N}$ must spike first. Then $\mathbf{s}|_{\mathcal{N}\setminus\{i\}}$ is a pure sequence on $n-1$ neurons. By induction on n, we have that $G(B_{\text{skew}}(\mathbf{s}|_{\mathcal{N}\setminus\{i\}}))$ is a transitive tournament. Because i precedes all other neurons, we have that $\beta_{ij}(\mathbf{s}) = 1$ for each $j \in \mathcal{N} \setminus \{i\}$. In particular, this means that the graph $G(B_{\text{skew}}(\mathbf{s})) = A$ is transitive, completing the proof. □

As a result of Lemma 3.30, it is clear that all possible transitive tournaments can be realized as the bias network of some sequence. But it is not obvious which other digraphs are realizable, and it would be surprising if a sequence could be constructed

to match an arbitrary list of pairwise orderings (as a simple digraph specifies).

Question 3.31. Which digraphs are realizable as the bias network of some sequence?

The following examples explore this question.

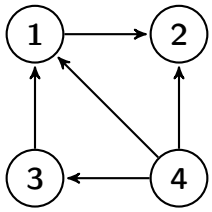
Example 3.32. The sequence $\mathbf{s} = 12444433131124$ has skew-bias matrix

$$\begin{bmatrix} 0 & 1/4 & -1/3 & -1/5 \\ -1/4 & 0 & 0 & -1/5 \\ 1/3 & 0 & 0 & -3/5 \\ 1/5 & 1/5 & 3/5 & 0 \end{bmatrix}.$$

This allows us to easily see that $G(\mathbf{s})$ has adjacency matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

Graphically, we can visualize $G(\mathbf{s})$ as below.

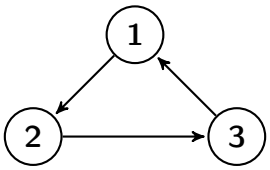


Notice that $G(\mathbf{s})$ is not a tournament because it does not contain an edge between vertices 2 and 3, which happened because $\beta_{23}(\mathbf{s}) = \beta_{32}(\mathbf{s}) = 0$.

Example 3.33. The sequence $\mathbf{s} = 2331112123$ has bias network $G(\mathbf{s})$ with adjacency matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Graphically, $G(\mathbf{s})$ is shown below.



So $G(\mathbf{s})$ is clearly not transitive since $(1,2),(2,3) \in E(\mathbf{s})$ but $(1,3) \notin E(\mathbf{s})$.

The answer to Question 3.31 is surprising: We have control over each individual tendency in a bias network.

Theorem 3.34. *Every bias network is possible.*

To prove this theorem, we first need some other results and definitions.

Definition 3.35. The *reverse* of a sequence $\mathbf{s} = (s_1, \ldots, s_\ell)$ is the sequence $\mathbf{s}^T = (s_\ell, s_{\ell-1}, \ldots, s_1)$.

Note that the reverse of the reverse of a sequence is the original sequence; that is $(\mathbf{s}^T)^T = \mathbf{s}$ for any sequence $\mathbf{s} \in \mathcal{S}$. The reason for the notation $\mathbf{s}^T$ is seen in the following lemma.

Lemma 3.36. *For any sequence* $\mathbf{s}$, $B_{skew}(\mathbf{s}^T) = B_{skew}(\mathbf{s})^T$.

Proof. Consider neurons $i, j \in \mathcal{N}$. For each index pair $(k,k') \in L_{ij}(\mathbf{s})$, we must have that $(k',k) \in L_{ji}(\mathbf{s}^T)$. Similarly, for each index pair $(m,m') \in L_{ij}(\mathbf{s}^T)$, we must have that $(m',m) \in L_{ji}(\mathbf{s})$. Hence, $c_{ij}(\mathbf{s}) = |L_{ij}(\mathbf{s})| = \left|L_{ji}(\mathbf{s}^T)\right| = c_{ji}(\mathbf{s}^T)$ for arbitrary neuron pairs. Now,

$$\beta_{ji}(\mathbf{s}^T) = \frac{c_{ji}(\mathbf{s}^T) - c_{ij}(\mathbf{s}^T)}{c_{ji}(\mathbf{s}^T) + c_{ij}(\mathbf{s}^T)} = \frac{c_{ij}(\mathbf{s}) - c_{ji}(\mathbf{s})}{c_{ij}(\mathbf{s}) + c_{ji}(\mathbf{s})} = \beta_{ij}(\mathbf{s}) = -\beta_{ji}(\mathbf{s}).$$

In particular, $B_{\text{skew}}(\mathbf{s}^T) = -B_{\text{skew}}(\mathbf{s}) = B_{\text{skew}}(\mathbf{s})^T$ since B_{skew} is skew-symmetric, thus finishing the proof. □

Lemma 3.37. *Given a sequence* $\mathbf{s}$*, a sign value* $a \in \{-1, 1\}$*, and distinct neurons* $i, j \in \mathcal{N}$*, there is a sequence* $\mathbf{s}'$ *such that*

1. $\operatorname{sign}(\beta_{ij}(\mathbf{s}')) = a$ *and*
2. $\operatorname{sign}(\beta_{qr}(\mathbf{s}')) = \operatorname{sign}(\beta_{qr}(\mathbf{s}))$ *for* $\{q, r\} \neq \{i, j\}$.

Proof. Recall that, for any distinct neurons $i, j \in \mathcal{N}$, $\beta_{ij} = \frac{c_{ij} - c_{ji}}{c_{ij} + c_{ji}}$. Since $c_{ij} + c_{ji}$ is a count of neuron pairs, it is positive; in particular, this tells us that $\operatorname{sign}(\beta_{ij}) = \operatorname{sign}(c_{ij} - c_{ji})$. Now, let $\mathbf{s} \in \mathcal{S}$ be given and fix $i, j \in \mathcal{N}$ as distinct neurons.

Since B_{skew} is skew-symmetric, assume without loss of generality that $a = 1$ (by exchanging i and j if $a = -1$). Pick $k \in \mathbb{N}$ such that $2k^2 > c_{ji}(\mathbf{s}) - c_{ij}(\mathbf{s})$, and define $\mathbf{s}' := i^k \cdot j^k \cdot \mathbf{s} \cdot i^k \cdot j^k$. We will first show that $\beta_{ij}(\mathbf{s}') > 0$, implying that $\operatorname{sign}(\beta_{ij}(\mathbf{s}')) = 1 = a$. Note that $c_{ij}(\mathbf{s}') = 3k^2 + k \cdot c_j(\mathbf{s}) + c_i(\mathbf{s}) \cdot k + c_{ij}(\mathbf{s})$ and that $c_{ji}(\mathbf{s}') = k^2 + k \cdot c_i(\mathbf{s}) + c_j(\mathbf{s}) \cdot k + c_{ji}(\mathbf{s})$. Now, $c_{ij}(\mathbf{s}) - c_{ji}(\mathbf{s}) = 2k^2 + c_{ij}(\mathbf{s}) - c_{ji}(\mathbf{s}) > 0$, as claimed.

Let $q, r \in \mathcal{N}$ be distinct with $\{q, r\} \neq \{i, j\}$. Suppose that $\{q, r\} \cap \{i, j\} = \emptyset$. Then $\beta_{qr}(\mathbf{s}') = \beta_{qr}(\mathbf{s}'|_{\mathcal{N} \setminus \{i,j\}}) = \beta_{qr}(\mathbf{s}|_{\mathcal{N} \setminus \{i,j\}}) = \beta_{qr}(\mathbf{s})$. In particular, $\operatorname{sign}(\beta_{qr}(\mathbf{s}')) = \operatorname{sign}(\beta_{qr}(\mathbf{s}))$. Now, suppose that $q \in \{i, j\}$ but $r \notin \{i, j\}$. Then $c_{qr}(\mathbf{s}') = k \cdot c_r(\mathbf{s}) + c_{qr}(\mathbf{s})$ and $c_{rq}(\mathbf{s}') = c_{rq}(\mathbf{s}) + c_r(\mathbf{s}) \cdot k$; hence, $c_{qr}(\mathbf{s}') - c_{rq}(\mathbf{s}') = c_{qr}(\mathbf{s}) - c_{rq}(\mathbf{s})$. In particular, $\operatorname{sign}(\beta_{qr}(\mathbf{s}')) = \operatorname{sign}(c_{qr}(\mathbf{s}') - c_{rq}(\mathbf{s}')) = \operatorname{sign}(c_{qr}(\mathbf{s}) - c_{rq}(\mathbf{s})) = \operatorname{sign}(\beta_{qr}(\mathbf{s}))$. Therefore, the lemma holds. □

Definition 3.38. A sequence $\mathbf{s}$ is a *palindrome* if $\mathbf{s}^T = \mathbf{s}$.

Lemma 3.39. *If* $\mathbf{s} \in \mathcal{S}([n])$ *is a palindrome, then* $B_{skew}(\mathbf{s}) = \mathbf{0}_n$*, where* $\mathbf{0}_n$ *is the* $n \times n$ *zero matrix.*

Proof. Since $\mathbf{s}$ is a palindrome, we have that $\mathbf{s} = \mathbf{s}^T$. Using Lemma 3.36 and the fact that the skew-bias matrix is skew-symmetric, we have that $B_{\text{skew}}(\mathbf{s}) = B_{\text{skew}}(\mathbf{s}^T) =$

$B_{\text{skew}}(\mathbf{s})^T = -B_{\text{skew}}(\mathbf{s})$. In particular, $\beta_{ij}(\mathbf{s}) = -\beta_{ij}$ for each $i, j \in \mathcal{N}$, implying that $\beta_{ij} = 0$. This completes the proof. □

We are now ready to show that every simple digraph is realizable as a bias network.

Proof of Theorem 3.34. Let D be an arbitrary digraph. Then D has skew-symmetric adjacency matrix $A \in \{-1, 0, 1\}^{n \times n}$. We want to show that there exists a sequence $\mathbf{s}$ on neuron set $\mathcal{N} = [n]$ such that $B_{\text{skew}}(\mathbf{s}) = A$.

Let $\mathbf{s}_0$ be a palindrome with $\text{supp}(\mathbf{s}_0) = \mathcal{N}$. By Lemma 3.39, we know that $B_{\text{skew}}(\mathbf{s}_0) = \mathbf{0}_n$ is the $n \times n$ zero matrix. Define $I := \{(i, j) \in \mathcal{N}^s \mid a_{ij} > 0\}$ to be the set of index pairs at which A is positive, and let $I = \{(i_1, j_1), \ldots, (i_m, j_m)\}$ be an enumeration of I. We proceed by induction on m. If $m = 1$, then A has only one positive entry $a_{i_1 j_1}$. Since $\mathbf{s}_0$ is the zero matrix, Lemma 3.37 provides a sequence $\mathbf{s}_1$ such that $\beta_{i_1 j_1}(\mathbf{s}_1) = 1$ and $\beta_{qr}(\mathbf{s}_1) = 0$ for all $q, r \in \mathcal{N}$ with $\{q, r\} \neq \{i, j\}$; that is $B_{\text{skew}}(\mathbf{s}_1) = A$.

Suppose now that $m > 1$, and assume for induction that there is a sequence $\mathbf{s}_{m-1}$ such that $\beta_{i_m j_m}(\mathbf{s}_{m-1}) = \beta_{j_m, i_m}(\mathbf{s}_{m-1}) = 0$ and $\beta_{qr}(\mathbf{s}_{m-1}) = a_{qr}$ for all other $q, r \in \mathcal{N}$. Lemma 3.37 now provides a sequence $\mathbf{s}_m$ such that $\beta_{i_m j_m}(\mathbf{s}_m) = 1$ and $\beta_{qr}(\mathbf{s}_m) = \beta_{qr}(\mathbf{s}_{m-1})$ for all other $q, r \in \mathcal{N}$. Then $B_{\text{skew}}(\mathbf{s}_m) = A$. Hence, by induction, the lemma holds for any $m > 0$. □

Chapter 4

Comparing sequences

4.1 Correlation between sequences

4 In the brain, it is often the case that signals are very noisy (due to neuronal misfires, improper spike sorting, etc.); as a result, comparing sequences directly to each other may not be so effective. Additionally, complications arise when one wants, for instance, to compare relatively long neuronal sequences (e.g., place-field sequences) to relatively short neuronal sequences (e.g., SWR sequences): How does one meaningfully (and generically) compare a sequence with 10 spikes to a sequence with 100 spikes? Even if a meaningful comparison were developed to directly compare the sequences, this comparison would likely be computationally expensive since these techniques usually involve modifying one sequence until it becomes the other sequence. As we attempt to compare sequences in a less direct way, we must ask a natural question: What does it mean for two sequences to be similar (or even "the same")? As we will hopefully show, the bias matrix provides a somewhat natural framework for answering this question and, thus, for comparing sequences.

Recall that the bias matrix arose out of an attempt to generalize our intuitive

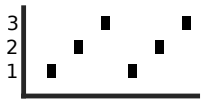


Figure 4.1: Spike raster plot of an impure sequence that induces an unambiguous neuronal ordering

notion of a pure neuronal sequence. Even with this representation of sequences as biases (which are vectors), careful consideration should be given to what method is employed to compare these vectors. It is tempting to want to compare biases using the standard Euclidean distance, but this may not be the most meaningful of comparisons. To see this, once again consider pure sequences. Recalling Lemma 3.6, the bias matrix of a pure sequence contains sufficient information to determine the original neuronal ordering. Pure sequences, though, are not the only sequences for which the bias matrix allows us to unambiguously reconstruct an ordering in the same manner. Indeed, some of the sequences corresponding to these other bias matrices even match up well with our original intuition.

Example 4.1. Consider the spike-raster plot in Figure 4.1. Though the spikes of the neurons are interlaced, it seems obvious to the casual observer that there is only one neuronal ordering that makes sense: neuron 1 fires, then neuron 2, and finally neuron 3. This casual observation seems slightly less obvious when considering the sequence representation $\mathbf{s} = 123123$, but it emerges again upon computing the bias matrix:

$$B_{\text{skew}}(\mathbf{s}) = \begin{bmatrix} 0 & 1/2 & 1/2 \\ -1/2 & 0 & 1/2 \\ -1/2 & -1/2 & 0 \end{bmatrix} = \frac{1}{2} B_{\text{skew}}(123).$$

The skew-bias matrix of $\mathbf{s}$ is a scaling of that of the pure sequence 123. As such, $B_{\text{skew}}(\mathbf{s})$ induces the same ordering as $B_{\text{skew}}(123)$, namely the ordering $(1, 2, 3)$.

From this perspective, it seems that the underlying "ordering," if you will, of a

bias matrix is characterized simply by the direction in which the bias matrix points (when considered as a vector). This further suggests that the magnitude of a bias vector is not so much an indication of an underlying order as it is an indication of the purity of a sequence. It is with this perspective that we define a correlation between sequences.

Definition 4.2. We define the *product* of sequences $\mathbf{s}$ and $\mathbf{s}'$ to be $\langle \mathbf{s}, \mathbf{s}' \rangle := \sum_{i,j \in \omega} \beta_{ij}(\mathbf{s}) \beta_{ij}(\mathbf{s}')$ for $\omega = \text{supp}(\mathbf{s}) \cap \text{supp}(\mathbf{s}')$. Note that $\langle \mathbf{s}, \mathbf{s}' \rangle = \langle \mathbf{s}|_\omega, \mathbf{s}'|_\omega \rangle = \langle B_{\text{skew}}(\mathbf{s}|_\omega), B_{\text{skew}}(\mathbf{s}'|_\omega) \rangle$, where the last expression is the standard Euclidean inner product of $B_{\text{skew}}(\mathbf{s}|_\omega)$ and $B_{\text{skew}}(\mathbf{s}'|_\omega)$. Also, we define the *magnitude* of a sequence $\mathbf{s}$ to be $\|\mathbf{s}\| := \sqrt{\langle \mathbf{s}, \mathbf{s} \rangle}$. With these expressions in hand, we can finally define the *correlation of sequences* $\mathbf{s}$ *and* $\mathbf{s}'$ to be

$$\text{corr}(\mathbf{s}, \mathbf{s}') := \frac{\langle \mathbf{s}, \mathbf{s}' \rangle}{\left\| \mathbf{s}|_{\text{supp}(\mathbf{s}')} \right\| \cdot \left\| \mathbf{s}'|_{\text{supp}(\mathbf{s})} \right\|}$$

In terms of geometry, $\text{corr}(\mathbf{s}, \mathbf{s}')$ is the cosine of the angle between $B_{\text{skew}}(\mathbf{s}|_\omega)$ and $B_{\text{skew}}(\mathbf{s}'|_\omega)$, where again $\omega = \text{supp}(\mathbf{s}) \cap \text{supp}(\mathbf{s}')$. In these definitions, we restrict to the common set of neurons in part because noise is a major consideration in real data. A real sequence might contain a spike from a neuron that should not have been active but not contain a spike from a neuron that should have been active. Though we cannot know for sure which spikes "should" and "should not" be in a sequence, we attempt to mitigate this problem slightly by comparing sequences based only on the collection of neurons that are active in both sequences.

Example 4.3. Let $\mathbf{s} = 12444433131124$ and $\mathbf{s}' = 2431$. Then

$$B_{\text{skew}}(\mathbf{s}) = \begin{bmatrix} 0 & 1/4 & -1/3 & -1/5 \\ -1/4 & 0 & 0 & -1/5 \\ 1/3 & 0 & 0 & -3/5 \\ 1/5 & 1/5 & 3/5 & 0 \end{bmatrix} \quad \text{and} \quad B_{\text{skew}}(\mathbf{s}') = \begin{bmatrix} 0 & -1 & -1 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \\ 1 & -1 & 1 & 0 \end{bmatrix}.$$

Noting that $\text{supp}(\mathbf{s}) = \text{supp}(\mathbf{s}') = [4]$, the correlation between $\mathbf{s}$ and $\mathbf{s}'$ is $\text{corr}(\mathbf{s}, \mathbf{s}') = \frac{\langle \mathbf{s}, \mathbf{s}' \rangle}{\|\mathbf{s}\| \cdot \|\mathbf{s}'\|}$. Now, each component of this can be computed as follows:

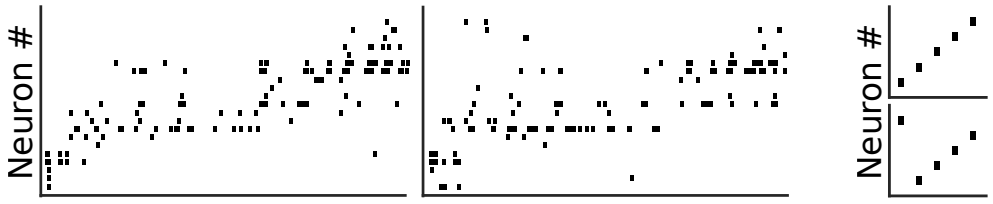
$$\begin{aligned}
\langle \mathbf{s}, \mathbf{s}' \rangle &= \sum_{i,j \in \mathcal{N}} \beta_{ij}(\mathbf{s}) \beta_{ij}(\mathbf{s}') \\
&= 2 \sum_{i=1}^{4-1} \sum_{j=i+1}^{4} \beta_{ij}(\mathbf{s}) \beta_{ij}(\mathbf{s}') \\
&= 2 \left(-\tfrac{1}{4} + \tfrac{1}{3} + \tfrac{1}{5} + 0 - \tfrac{1}{5} + \tfrac{3}{5} \right) \\
&= \frac{41}{60} \\
\|\mathbf{s}\| &= \sqrt{2 \left(\tfrac{1}{16} + \tfrac{1}{9} + \tfrac{1}{25} + 0 - \tfrac{1}{25} + \tfrac{9}{25} \right)} = \frac{47\sqrt{2}}{60} \\
\|\mathbf{s}'\| &= \sqrt{12} = 2\sqrt{3}
\end{aligned}$$

Putting this all together, we find that $\text{corr}(\mathbf{s}, \mathbf{s}') = \frac{41}{47\sqrt{6}} \approx 0.6168$.

4.2 Significance of correlation

While we do now have the ability to compare arbitrary sequences using our correlation measure, there is still an issue to be addressed: How do we interpret a correlation value? Despite coming to the bias-matrix representation of a sequence and the correlation measure by fairly intuitive means, a correlation value does not seem to have an intuitive interpretation beyond (1) its sign indicating correlation or anti-correlation and (2) larger magnitude indicating strength of (anti-)correlation.

In Figure 4.2, we see an example of how a correlation value alone does not necessarily say so much about the similarity of two sequences. Why does something like this happen? This is partly an issue of angles and dimensionality. Consider the pure sequences in Figure 4.2b. By moving just a single spike, we modified a large



(a) Arm-run (b) Pure

Figure 4.2: Both of these pairs of sequences seem to be similar (to the eye), so we might expect them to have relatively high correlation values. However, neither pair does: The arm-run sequences have a correlation value of 0.52572, and the pure sequences have a correlation value of 0.2. These low values are somewhat unsatisfactory since maximally uncorrelated sequences have a correlation value of zero.

proportion of the entries in the pairwise-bias matrix. If these short sequences had a common subsequences prepended to them, then the correlation value would be much higher even though the same number of spikes are "out of order"; this is because a much larger percentage of the pairwise-bias matrix entries are unmodified in this case. When considering the arm-run sequences in Figure 4.2a, we notice that each of the neurons has spikes that are interlaced with spikes from many of the other neurons. When spikes are interlaced in a consistent way like in Example 4.1, bias matrices end up being scalings of pure biases. These spikes, however, are not interlaced in such consistent ways; and the result is that the bias matrices end up pointing in fairly different directions. Still, to the eye, these long arm-run sequences could be much more different from each other if we just shifted some of their spikes around.

And here we stumble onto some key intuition: We evaluate the similarity of one sequence $\hat{\mathbf{s}}$ to another $\mathbf{s}$ not just on how close $\hat{\mathbf{s}}$ is to $\mathbf{s}$ but also on how much more different they could be from each other. In a sense, we ask the question of where their similarity falls in relation to the similarities of other pairs of sequences.

Definition 4.4. Given sequences $\mathbf{s}$ and $\hat{\mathbf{s}}$, the *significance of correlation* of $\hat{\mathbf{s}}$ to $\mathbf{s}$ is the proportion of shufflings of $\hat{\mathbf{s}}$ that are more strongly correlated to $\mathbf{s}$ than $\hat{\mathbf{s}}$ is to

$\mathbf{s}$. More precisely, letting $\Pi = \{\pi \in \text{Perms}([\ell']) : |\text{corr}(\mathbf{s}, \hat{\mathbf{s}})| \leq |\text{corr}(\mathbf{s}, \hat{\mathbf{s}}^\pi)|\}$, then the significance of the correlation of $\hat{\mathbf{s}}$ to $\mathbf{s}$ is $\text{sig}_{\mathbf{s}}(\hat{\mathbf{s}}) := \frac{|\Pi|}{\text{len}(\hat{\mathbf{s}})!}$.

Example 4.5. We used computational techniques (as will be described in Section 4.4) to approximate the significance values corresponding to the correlations of Figure 4.2. In our simulations, the significance of the correlation values of the arm-run sequences and of the pure sequences were approximately 0.00 and 0.40, respectively. Using a standard \$p\$-value of 0.05, This indicates that the correlation value between the arm-run sequences is significant but that the correlation value between the pure sequences is not significant.

4.3 Global significance

The previous section allows us to compare collections of sequences while putting each correlation on more even footing; i.e., we can set a threshold for significance instead of for (absolute value of) correlation. Now, we are brought to the question of similarity of one collection of sequences to another. Let $U = (\mathbf{u}_1, \ldots, \mathbf{u}_a)$ and $V = (\mathbf{v}_1, \ldots, \mathbf{v}_b)$ be lists of sequences in $\mathcal{S}$. We want to ask how similar is U to V. One natural place to start answering this question is to consider all of the correlation values between a sequence in U and a sequence in V.

Definition 4.6. We define the *correlation matrix* between U and V to be $\text{Corr}(U, V) := [\text{corr}(\mathbf{u}, \mathbf{v})]_{\mathbf{u} \in U,\ \mathbf{v} \in V}$.

Though we can look for commonalities and statistics among the entries of this matrix (e.g., that it has all positive entries or by considering the distribution of values), we face a similar problem as when looking at individual correlation values: Aside from basic properties (such as having only positive entries), we do not have a

good interpretation of the magnitude of correlation (as demonstrated in the examples of the previous section). As such, we will approach the answer to this qeustion in the same way as for individual correlations: by considering significance values. But we will need to iterate the process, in a sense, because we are comparing collections instead of just individual sequences.

Definition 4.7. Define the *significance matrix* of correlation of U to V to be $\mathrm{Sig}_V(U) := [\mathrm{sig}_{\mathbf{v}}(\mathbf{u})]_{\mathbf{u}\in U,\ \mathbf{v}\in V}$.

The significance matrix allows us to identify how significantly correlated each sequence in U is to each sequence in V. Still, though, the question remains: How do we convert this information to some sort of meaningful value to tell us how similar U is to V? Given a significance parameter $p \in (0,1)$, we can count how many pairs $(\mathbf{u},\mathbf{v}) \in U \times V$ satisfy $\mathrm{sig}_{\mathbf{v}}(\mathbf{u}) < p$. Though this gives us, for instance, the percentage of pairs that are significantly correlated (as determined by p), we have still not quite reached a satisfactory measure. What percentage of pairs is a significant percentage? It really depends on the collections U and V. As we compared $\mathrm{corr}(\mathbf{s},\hat{\mathbf{s}})$ to a distribution in the previous section, we must find an appropriate distribution of values to which we can compare the count $\mathrm{sig}_V(U,p) := |\{(\mathbf{u},\mathbf{v}) \in U \times V \mid \mathrm{sig}_{\mathbf{v}}(\mathbf{u}) < p\}|$. Naturally, as before, we want to compare this value to the counts gotten by comparing a random shuffling of the sequenes in U to V. If $\vec{\pi} = (\pi_1,\ldots,\pi_a) \in \Pi(U) := \mathrm{Perms}([\mathrm{len}(\mathbf{u}_1)]) \times \cdots \times \mathrm{Perms}([\mathrm{len}(\mathbf{u}_a)])$, define $U^{\vec{\pi}} := (\mathbf{u}_1^{\pi_1},\ldots,\mathbf{u}_a^{\pi_a})$. Now, the *global significance of correlation* of U to V is $g_V(U,p) = \frac{|A|}{|\Pi(U)|}$ where $A = \left\{\vec{\pi} \in \Pi(U) \mid \mathrm{sig}_V(U^{\vec{\pi}},p) \geq \mathrm{sig}_V(U,p)\right\}$. Hence, we have a measure telling us how significantly correlated the list U is to the list V.

4.4 Monte-Carlo computation of significance

Mathematically, the significance of correlation and global significance of correlation values described in the previous two sections allow us to have confidence that two sequences or two collections of sequences share (or do not share) similarities. In practice, however, the distributions necessary for computation of these values are often prohibitively large. For instance, if $\hat{\mathbf{s}}$ is a sequence with only ten spikes, calculation of $\text{sig}_{\mathbf{s}}(\hat{\mathbf{s}})$ requires computation of over 3.6 million correlation values. And the situation gets a lot worse as the length of $\hat{\mathbf{s}}$ grows, not to mention the added complexities when trying to compute a global significance value.

Because of this difficulty, we employ a Monte-Carlo method to approximate the two types of significance values. In the analysis that follows in Section 5.3, each sequence is a part of some collection of sequences. We leverage this fact by simultaneously approximating the distribution for both types of significance values. Let T be some (large) natural number representing the number of trials in our Monte-Carlo approximations. Further, let $U = (\mathbf{u}_1, \ldots, \mathbf{u}_a)$ and V be lists of sequences; and let $p \in (0, 1)$ be a significance parameter. For each $\mathbf{u} \in U$ and each $\mathbf{v} \in V$, we will approximate $\text{sig}_{\mathbf{v}}(\mathbf{u})$ in such a way that the resultant distribution approximations of all such pairings can also be used to approximate the distributions necessary for computing $g_V(U, p)$.

Let $\vec{\pi}_1, \ldots, \vec{\pi}_T \in \Pi(U)$ be T randomly selected lists of permutations. For each $t \in [T]$, we compute and store the correlation matrix $\text{Corr}(V, U^{\vec{\pi}_k})$. This list of correlation matrices will allow us to approximate all significance values. First, to compute $\text{sig}_{\mathbf{v}}(\mathbf{u}_m)$, note that the list $\text{corr}(\mathbf{v}, \mathbf{u}_m^{\pi_{1,m}}), \ldots, \text{corr}(\mathbf{v}, \mathbf{u}_m^{\pi_{T,m}})$ can be extracted from the list of correlation matrices; hence, we approximate $\text{sig}_{\mathbf{v}}(\mathbf{u}_m)$ by $\widehat{\text{sig}}_{\mathbf{v}}(\mathbf{u}_m) = \frac{|A|}{T}$ where $A = \left\{t \in [T] : |\,\text{corr}(\mathbf{v}, \mathbf{u}_m)| < |\,\text{corr}(\mathbf{v}, \mathbf{u}_m^{\pi_{t,m}})|\right\}$. Indeed, this process produces

an approximation $\widehat{\mathrm{Sig}}_V(U)$ of the significance matrix $\mathrm{Sig}_V(U)$.

To approximate $g_V(U)$ using the list of computed correlation matrices, we note first that the list $U^{\vec{\pi}_k}$ is a shuffling not only of the list U but also of the list $U^{\vec{\pi}_{t'}}$ for each $t' \in [T] \setminus \{t\}$. As such, we can also approximate the significance matrices $\mathrm{Sig}_V(U^{\vec{\pi}_t})$ by $\widehat{\mathrm{Sig}}_V(U^{\vec{\pi}_t}) = [\widehat{\mathrm{sig}}_{\mathbf{v}}(\mathbf{u}_m^{\pi_{t,m}})]_{\mathbf{v}\in V, m\in[a]}$ with $\widehat{\mathrm{sig}}_{\mathbf{v}}(\mathbf{u}_m^{\pi_{t,m}}) = \frac{|A|}{T}$ where $A = \left\{t' \in [T] : |\operatorname{corr}(\mathbf{v}, \mathbf{u}_m^{\pi_{t,m}})| < |\operatorname{corr}(\mathbf{v}, \mathbf{u}_m^{\pi_{t',m}})|\right\}$. With these matrices in hand, the approximation of $g_V(U,p)$ is straightforward: Let $\widehat{\mathrm{sig}}_V(U',p) = \left|\left\{(\mathbf{u},\mathbf{v}) \in U \times V \mid \widehat{\mathrm{sig}}_V(U',p)\right\}\right|$, from which we define $\hat{g}_V(U,p) = \frac{|A|}{T}$ where $A = \left\{t \in [T] \mid \widehat{\mathrm{sig}}_V(U^{\vec{\pi}_t},p) \geq \widehat{\mathrm{sig}}_V(U,p)\right\}$.

Chapter 5

Experimental set-up and data analysis

The tools introduced in the preceding chapters were developed for use in the analysis of real-world data. The lab of Eva Pastalkova at the Howard Hughes Medical Institute's Janelia Research Campus graciously provided all of the data used in this document. The following section describes the experimental setup from which this data was collected and the various types of neuronal sequences that are considered in our analysis. Section 5.2 then describes details of our sequence detection techniques. Finally, Section 5.3 uses bias matrices to perform some basic analyses of the data.

5.1 The experimental setup and event types

Rats were trained to perform a basic memory task in the U-shaped track pictured in Figure 5.1. In short, the task of the animal can be described as follows: To receive a reward, the animal must alternate between running to the left reward port and the right reward port. The task is complicated for the animal by the delay area, which

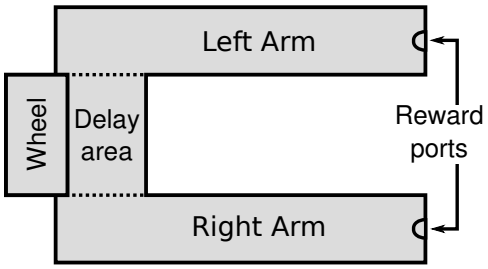


Figure 5.1: Alternation memory task. The three main areas of interest for this task are (1) the delay area and running wheel, (2) the left and right arms/corridors, and (3) the reward area at the end of each arm. The dotted lines surrounding the delay area represent mechanically movable doors that are used to hold the animals in the delay area until the wheel-run portion of each trial has been completed.

forces the animal to run in the running wheel for a specified amount of time before the doors of the delay area open and allow the animal to proceed toward a reward port. Each trial of the experiment consists of a run in the wheel, a visit to a reward port, and a return to the delay area. By requiring the animal to spend time running in the wheel before choosing to run to the left or right, the animal is forced to remember the arm to which it should next run instead of being allowed to run immediately from one arm to the other.

In addition to performing this task under normal conditions, each animal performed the task after the drug muscimol had been injected into the animal's medial septum. This injection has the effect of shutting down the medial septum, which is the area of the brain that serves as the primary drive of the theta oscillations that are thought to be an integral part of the generation of place-field sequences in the hippocampus during locomotion.

During performance of this task, the rat's location within the track was recorded. Further, local-field potentials (LFPs) were recorded at various depths across the pyramidal layer of the CA1 region of the rat's hippocampus. Neuronal spiking information was then extracted from these LFPs via a process called spike sorting. For our pur-

poses, we will say that an *event* is all of the information (LFPs, spike trains, animal location, etc.) within some time window. The sequence corresponding to an event is the list of all spikes that occur within that event. The various event/sequence types that we will be investigating are described below.

Wheel-run events are behavioral events recorded while the animal is running in the wheel to satisfy the delay requirement. As such, each wheel-run event is recorded over a period of approximately 10 seconds. These event types are generally the longest in duration of all of the event types that we consider; thus, the corresponding wheel-run sequences are usually the longest of our recorded sequence types. Because the animals are running during these sequences, the hippocampus exhibits a prominent theta oscillation under normal conditions.

Arm-run events are behavioral events recorded while an animal is running through either arm of the track in either direction between the delay area and a reward area. The duration of each arm-run event is the length of time that it takes the animal to get where it is going, usually around 2 seconds. Though shorter than wheel-run sequences, arm-run sequences are still usually quite long. The locomotion of the animal is again accompanied by a prominent theta oscillation. Because the animals are actually moving through the track during these sequences, the sequences exhibit place fields under normal conditions. We divide these sequences into four subtypes based on the trajectory of the animal; specifically, the animal can run to (outbound) or from (inbound) either the left or right reward area.

Sharp-wave ripple (SWR) events usually occur while an animal is not in motion; hence, they primarily occur in the reward areas and in the delay area outside of the wheel, though we placed no explicit restrictions on the location or move-

ment of the animal in our detection of these events. As described in Section 1.1, these events are characterized by a hyper-polarization in the LFP across the pyramidal layer in the CA1 region of the hippocampus, which generally lasts from around 50 to 250 milliseconds. Given the shorter timescale than the behavioral events, these sequences are generally much shorter than wheel-run and arm-run sequences.

Refined SWR events are the bursts of neuronal activity that accompany SWRs. These bursts of activity are not always perfectly aligned with the corresponding sharp wave, which (as we will see in the next section) is the feature we use to detect the SWR events. Given that our analysis is performed on sequences of neuronal spikes, it is important that we strive to find the sequences of spikes that most accurately represent the neuronal activity that accompanies the SWR events, regardless of whether or not those sequences are aligned with the accompanying LFP envelope. This refinement process results in the discarding of many events that do not contain sufficient spiking information. Thus, there are fewer refined SWR sequences than (unrefined) SWR sequences; but the sequences that remain are often longer.

5.2 SWR detection and refinement

Unlike arm-run and wheel-run events, SWR events are not behavioral in nature. In particular, no action or activity of an animal will tell us when a SWR event is occurring. As such, we must find a way to determine when these events are occurring. To do this, we consider one prominent feature of SWR events: the sharp wave. During a sharp wave, the LFPs across the pyramidal layer of CA1 become hyperpolarized,

resulting in the characteristic shape that has already been observed (see Figure 1.2). Our method for detecting SWR events can be summarized as follows:

1. Select two LFP channels, a lower channel (i.e., one that drops in value during SWRs) and an upper channel (i.e., one that increases in value during SWRs). In lieu of an automated method for performing this, all of the LFP channels were selected by hand. The selection process was done by gaining an inuitive understanding of what SWR events look like and then by selecting the channels that most-faithfully represented the desired LFP separation.

2. Center each LFP at zero by computing a local average for each. This is necessary to account for changes in the LFPs that occur over a longer time scale than SWRs occur on. To do this, we computed the average output for each channel over the interval of one second preceding each point in time; we then subtracted the resultant local averages from the original signals.

3. Create a single sharp-wave signal by subtracting the lower centered-LFP from the upper centered-LFP, subtracting the overall mean of the resultant signal, normalizing by the standard deviation of the resultant signal, and smoothing the resultant signal using a gaussian filter.

4. Threshold the sharp-wave signal to determine initial estimates for SWR event windows. We set the base-line threshold at 1.25 standard deviations of the sharp-wave signal.

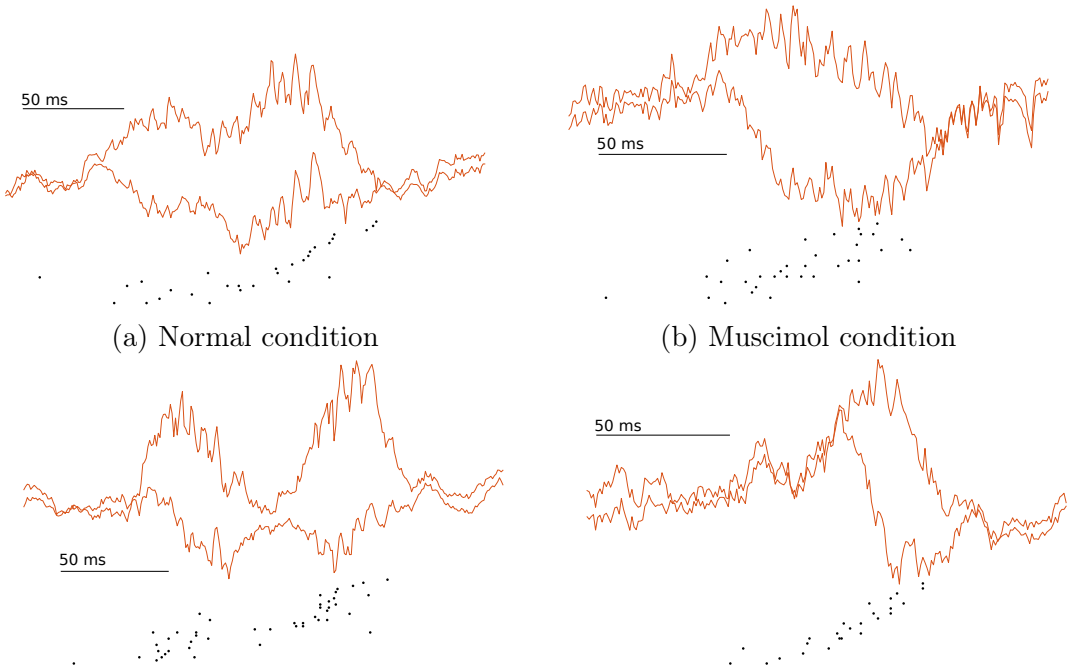
5. Filter, combine, and modify the initial event estimates to ensure that each event satisfies properties characteristic of SWRs.

    - Split single events into multiple events. Sometimes, SWRs happen close enough together that the separation in the LFP from the first event has

not ended before the separation from the second event has begun. As a result, the detection process up to this point joins those multiple events into a single event. This process of splitting is done by finding the peaks of the second derivative of the smoothed sharp-wave signal; if a large enough peak occurs within an event, then the event is split at that point in time.

- Remove events that do not last for some minimum amount of time (25 ms).
- Shorten events that last for too long. It is sometimes the case that the minimum-required LFP separation of a detected event can persist for longer than the underlying SWR event. To account for this while not discarding legitimate SWRs, we enforce a maximum SWR duration (250 ms) by taking the event in a 250-ms time window surrounding the highest point of LFP separation within the event.
- Remove events in which the peak of the sharp-wave signal is not large enough (4 standard deviations of the sharp-wave signal).
- Remove events that do not contain a sudden LFP separation. Without this characteristic feature, the event is not a SWR. We enforce this by ensuring that the absolute value of the first derivative of the sharp-wave signal passes above a minimum value.
- Ensure that enough spikes happen close enough to each other within each event. This is done by summing up gaussian functions centered at the spikes within each event and ensuring that the resultant signal passes above a minimum value.

The parameters used in this detection process were fine-tuned by hand until the resultant events were largely undeniably SWRs but also while not being so restrictive



(a) Normal condition (b) Muscimol condition

(c) Doublet (d) Needs refinement

Figure 5.2: SWR events. Panels (a) and (b) show standard examples of SWRs. Panel (c) shows an example of a doublet, two SWR events that occur so close in time as to be joined together. Panel (d) shows an example of a SWR for which the burst of neural activity significantly precedes the accompanying sharp wave. Such an event requires refinement.

as to eliminate many good candidates for SWR events.

Although detection based on the sharp wave gives us a good starting place for SWR detection, a detection algorithm based solely on the sharp wave is insufficient for our needs. While the bursts of neural activity in a SWR are (by definition) accompanied by a sharp wave, it is not the case that the two types of activity are perfectly synchronized in time. See Figure 5.2d for an example of this. Since our analysis is of neuronal sequences, it is crucial that we have as accurate a set of sequences as possible. Our attempt to extract the events corresponding to the neuronal-activity bursts of SWR events consists of four basic steps:

1. Detect the intrinsic sequences of a recording. These are formed by splitting the

spike train of a recording wherever the gap between successive spikes exceeds some threshold and then by extracting the sequences from each resultant spike train.

2. Discard all intrinsic sequences that do not overlap with any SWR event.

3. Join together those intrinsic sequences that overlap with a common SWR event. The idea is that the SWR event should represent a single pice of sequential information. Any gap that occurs within such a joining of intrinsic sequences is justified as being within a single sequence by arguing that other neurons—neurons that were not detected—were likely spiking within that interval.

4. Of the resulting collection of sequences, keep only those that meet certain requirements. First, we impose a requirement on the minimum number of active neurons; a sequence does not contain much useful information to us if, for instance, it contains spikes from only one neuron. Next, we impose a duration condition; there is reason to believe that SWRs have minimum and maximum duration. Lastly, we require that the sequence is isolated in time from its nearest intrinsic sequences; this is an attempt to ensure that the activity captured by the sequence is a burst of activity accompanying a SWR and not due to increased neural activity in the area for some other reason.

This process generally results in significantly fewer events than the sharp-wave detection process.

5.3 Sequence comparison in *in-vivo* hippocampus

It is believed that theta oscillations are necessary for the generation of hippocampal sequences. Analyses suggesting this result [10] compare observed hippocampal

sequences under normal and theta-inhibited (i.e., muscimol) conditions to neuronal templates generated, for instance, by considering the order of place fields in a track under normal conditions. The introduction of the bias matrix allows for the direct comparison of observed sequences to each other. In this section, we use the techniques of Chapter 4 to perform similarity analyses on the hippocampal sequences described in Section 5.1. In particular, we will attempt to determine whether the sequences generated under the influence of muscimol are different from those generated under normal conditions. For reference, brief descriptions of the sequence types are given in Table 5.1.

Table 5.1: Hippocampal sequence types.

Type	**Subtypes**	**Description**
Wheel-run		Behavioral sequences generated during the enforced delay period while running in the wheel
Arm-run	Left / outbound Left / inbound Right / outbound Right / inbound	Behavioral sequences generated as an animal traverses a path between the delay area and a reward area
Refined SWR		Non-behavioral sequences from SWRs

Question 5.1. Are hippocampal sequences the same after elimation of theta oscillation?

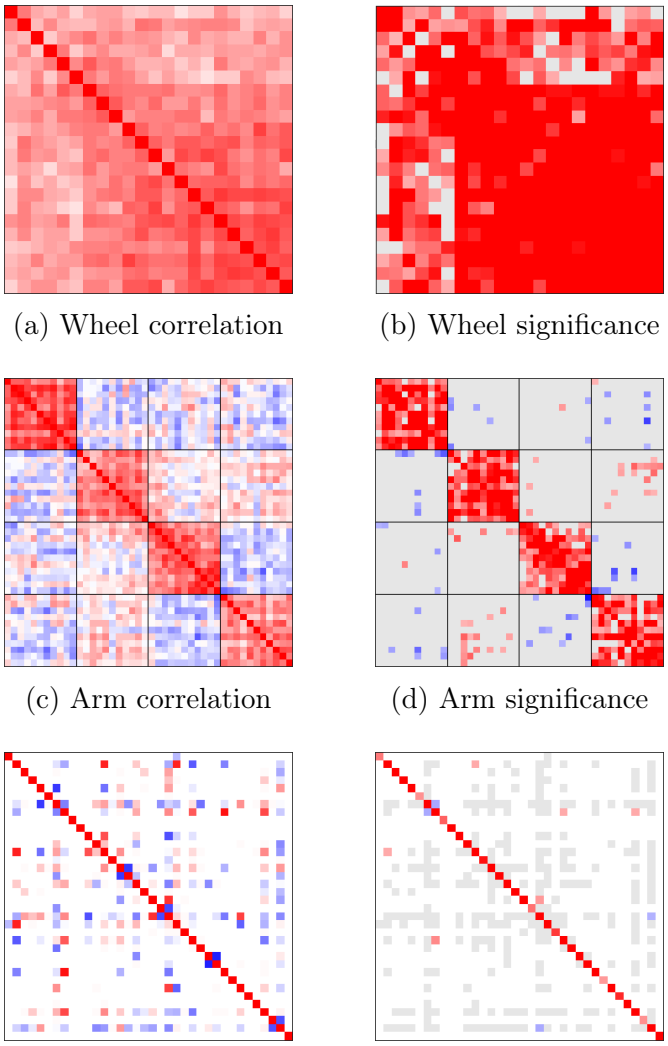
In the end, our analysis suggests that eliminating theta oscillation eliminates (at least some of) the structure of behavioral hippocampal sequences. We approach this question by setting *reference sets* of sequences to which we can compare normal-condition sequences and muscimol-condition sequences. Since the question we are trying to answer is about whether there is a change from the normal behavior, these reference sets should consist of normal-condition sequences. For instance, when comparing normal-condition wheel-run sequences to muscimol-condition wheel-run sequences, the reference set will be the collection of normal-condition wheel-run se-

quences. The basic tools used in this section are the correlation and significance matrices introduced in Section 4.3. Figure 5.3 provides an example of the correlation and significance matrices for each sequence type. The examples in this figure illustrate how the correlation matrix can be a useful tool for observing structure in collections sequences. In particular, notice the difference in the matrix structure between wheel, arm, and SWR correlations. Where the wheel-run correlation matrix displays that all correlations are positive, the arm-run correlation matrix shows positive and negative correlations existing in a highly structured manner (switching sign as we go between comparing subtypes of these sequences); the SWR correlation matrix, on the other hand, shows that positive and negative correlations exist in a much less structured way.

The differences in structure between the correlation matrices shown in Figure 5.3 can be observed over all data sets in Figure 5.4, which shows histograms of the correlation values for the different types. For instance, looking at the distributions of correlation values for arm-run sequences, the normal-condition correlations tend to be more positive than the muscimol-condition correlations. This observation is in line with the intended purpose of the muscimol injections to eliminate the theta oscillation that is believed to aid in the formation of place-field sequences.

In addition to looking for changes in the distributions of correlation values after injection of muscimol, we also investigate the change in proportion of correlations that are significant after injection of muscimol, as shown in Figure 5.5. For behavioral events, muscimol is accompanied by a reduction in the proportion of correlations that are significant. The reduction is somewhat modest for arm-run sequences when compared to the reduction for wheel-run sequences. This suggests that theta oscillation is more critical for the formation of normal-condition wheel-run sequences than for the formation of normal-condition arm-run sequences.

In conclusion, our analysis demonstrates via bias-matrix correlations that behavioral hippocampal sequences lose some of their structure when theta oscillation is removed by the injection of musimol into an animal's medial septum.

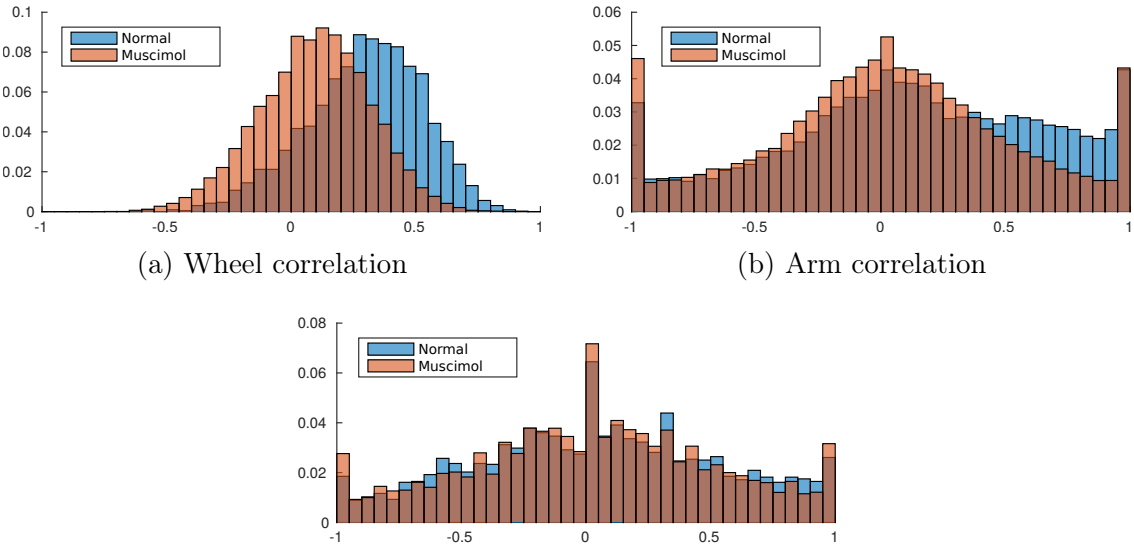


(a) Wheel correlation (b) Wheel significance

(c) Arm correlation (d) Arm significance

(e) SWR correlation (f) SWR significance

Figure 5.3: Correlation and significance matrices. Red entries indicate positive correlation, blue entries indicate negative correlation, and white entries indicate incomparable sequences; in significance matrices, gray entries indicate insignificant correlation values. Panels (a) and (b) show that all wheel sequences are positively correlated to each other and that a majority of the correlations are significant. Panels (c) and (d) show matrices for arm-run sequences; events are sorted by their subtypes: left/outbound, left/inbound, right/outbound, left/inbound. Lines delineate subtypes. The checkering pattern indicates postitive correlation within a subtype and negative correlations between inbound and outbound events. Few correlations between subtypes are significant. Panels (e) and (f) show the sparsity of correlation matrices for SWR sequences; additionally, few of the comparable SWR events are significantly correlated. The all-red diagonals indicate that each event is positively and significantly correlated to itself. All matrices were generated from the same recording.

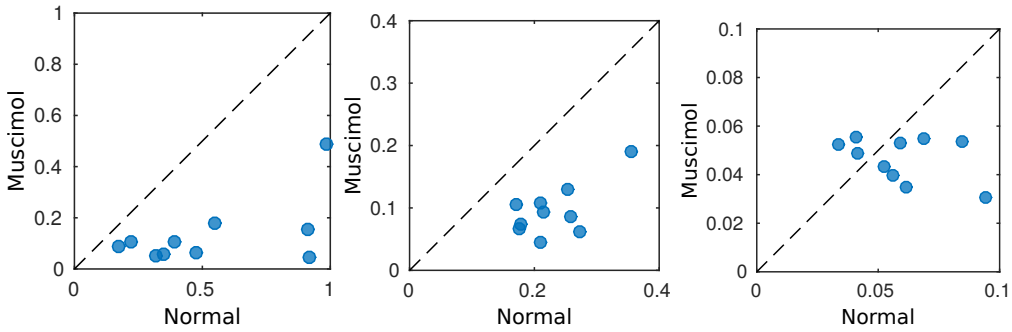


(a) Wheel correlation

(b) Arm correlation

(c) SWR correlation

Figure 5.4: Histograms of correlation values. For each type of sequence, the corresponding plot above shows the distribution of correlation values for normal-condition and muscimol-condition correlation values across all data sets. In panel (a), the wheel-run histograms show the greater tendency of normal-condition sequences to be positively correlated (88.75% positive) than their muscimol counterparts (67.86% positive). Panel (b) illustrates a similar phenomenon for normal-condition arm-run sequences to exhibit stonger positive correlations than their muscimol counterparts. Panel (c) shows a much less pronounced difference when considering SWR sequences, though the two-sample Kolmogorov-Smirnov test indicates that the distributions are indeed different (p-value of 0.0097).



(a) Wheel (b) Arm (c) SWR

Figure 5.5: Proportion of correlations that are significant. For each recording and each correlation/significance matrix, we consider the proportion of significance values that are below some threshold (0.05); this process results in two values, one for the normal-condition sequences and one for the muscimol-condition sequences, the horizontal and vertical axes, respectively, of the plots above. Panels (a) and (b) show that injection of muscimol is accompanied by a systematic drop in the proportion significant correlations for both sets of behavioral sequences. This universal drop in significance is not observed when considering SWR sequences in panel (c).

Appendix A

README for codebase

This code is designed to provide tools for working with neuronal sequences. The project started originally as a set of functions to detect sharp-wave ripples in the hippocampus. The code breaks down into three basic pieces:

- The `sequences` folder: At the core of this project, the aim was to analyze neuronal sequences. Our concept of a sequence is simply a list of spikes. Because we discard timing information from the spike trains, the techniques implemented here are agnostic of the timescale on which such a sequence was actually recorded.

- The `Event` class: An `Event` is a convenience class that was designed (1) to store a sequence and additional information about its origin in a recording and (2) to bundle together functions for working with sequences.

- The `NeuralData` class: A `NeuralData` object is our interface to the raw data from a neural recording. This class provides functions for retrieving recording information, extracting LFPs, and detecting various event types, among some other convenience functions.

The sections below describe each of these pieces in more detail along with some basic examples of how to use this project. This project is dependent on some additional functionality found in the matlab-incremented project.

A.1 Sequences

Since a sequence as we have it defined is just a list of spikes (i.e., neurons), each sequence represented in this project as a vector. As such, many MATLAB functions for vectors can come in handy. For instance, the collection of neurons that are active in a sequence `s` can be found with `unique(s)`.

```
s = [1, 2, 1, 3, 2, 2, 4, 3, 3, 4]
unique(s)
```

A few sequence-specific functions have been written where there was no equivalent for vectors. The spike-count vector can be computed with `spikecount`, and the index sets where neurons spike can be extracted with `spikesets`.

```
spikecount(s)
spikesets(s)
```

To visualize `s`, create a spike-raster plot.

```
spikeraster(s)
```

The center-of-mass ordering of a sequence `s` can be computed with `sortneurons(s)`.

```
sortneurons(s)
```

The bias-count and skew-bias matrices of `s` can be computed with `biascount(s)` and `skewbias(s)`, respectively. Both of these functions return a sparse matrix.

```
full(biascount(s))
full(skewbias(s))
```

The spikes of a sequence can be permuted to be in a random order using `shuffle`.

```
u = shuffle(s)
```

To compute the correlation between sequences, use `correlation`.

```
correlation(s, u)
```

For a list `L1` of sequences (i.e., a vector cell array), use `activity(L1)` to return a matrix in which row `i` is an indicator vector for the set of neurons that are active in `L{i}`.

```
randseq = @(~) randi(5, 1, 3);
L1 = arrayfcn(randseq, zeros(3, 1));
L1{:}
activity(L1)
```

With a second list `L2` of sequences, the number of neurons that each pair of sequences has in common can be found with `coactivity(L1, L2)`.

```
L2 = arrayfcn(randseq, zeros(3, 1));
L2{:}
coactivity(L1, L2)
```

A.2 Events

When dealing with sequences in real data, it is often necessary for purposes of analysis to know more about each sequence. In concept, an event is all of the information about

a recording that occurs within some time window; as such, the `Event` object class stores this time window. However, extracting information from a `NeuralData` object can be time-consuming; so this class also stores some additional information. The members of this class are listed below:

`window` This parameter stores the time window during which an event occurs. It is stored as a vector of the form `[start, end]`. For consistency regardless of the sampling rate of a recording, this is stored with units of seconds.

`spikes` This parameter stores the list of spikes that occur during an event. That is, `spikes` is the neuronal sequence contained within an `Event` object.

`times` This parameter stores the times corresponding to the event's spikes.

`type` This parameter allows the user to label each event as some specific type. A type can be an arbitrary string.

Though an `Event` can be created manually, it is often most useful in that several methods of the `NeuralData` class return (lists of) events.

A.3 Interfacing with real data

The code in the `NeuralData` class was originally designed to provide functionality for detecting sharp-wave ripples in LFP signals.

- TODO: The current setup requires a `.dat` file with a specific naming convention. It also requires a `*_BehavElecDataLFP.mat` file and some of its kin. This should be updated so that these extra files are not necessary. In the meantime, mention these sisues and direct the user to the github page. As an intermediate step, perhaps add a function for generating these files. Make the `.dat`

file optional if a cache directory exists. Use `.res` (etc.) files instead of the `*_BehavElecDataLFP.mat` file.

For the sake of example, suppose that the directory `A111-20150701` contains the files `A111-20150701.dat` and `A111-20150701_BehavElecDataLFP.mat`. Also, assume that `cache` is a directory. We can create a `NeuralData` object by specifying the data and cache directories.

```
nd = NeuralData('A111-20150701', 'cache');
```

We can now extract various kinds of information from the object `nd`. To detect SWR events, three channels need to be selected by hand using an external tool. Once the channels have been selected, load them with `loadChannels`.

```
loadChannels(nd)
```

The SWR events can now be detected with `detectRipples`.

```
cellRipples = detectRipples(nd);
```

The output of this method is a cell array of `Event` objects. These events can be visualized with `browseEvents`.

```
browseEvents(nd, cellRipples)
```

- TODO: The functions `browseEvents` and `browseSequences` should be modified/combined. The new `browseEvents` should have signature `browseEvents(cellEvents, <nd>)` and only plot the spike-raster plot if no `NeuralData` object is provided. The new `browseSequences` should be `browseSequences(cellSeqs) = browseEvents(cellfcn(@seq2evt, cellSeq))`.

The locations of the animal in the track at a particular list of times can be retrieved with the methods `getLocationsAtTimes`. For instance, we might want to know where ripples occurred in the track. To do this, we will need a single time to represent each event. Let's choose the starting time of the events.

```
vTimes = cellfun(@startTime, cellRipples);
mtxLocs = getLocationsAtTimes(nd, vTimes);
```

We can now visualize where the ripples occur on the track.

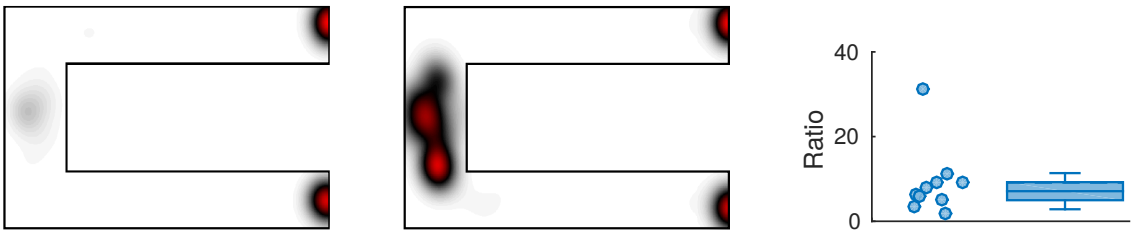
```
vWinX = minmax(mtxLocs(:, 1));
vWinY = minmax(mtxLocs(:, 2));
dStd = 0.1 * min(diff(vWinX), diff(vWinY));
nResolution = 1000;

imagesc(psth2d(mtxLocs, vWinX, vWinY, dStd, nResolution))
```

Many other `NeuralData` methods exist for extraction of recording information and event information.

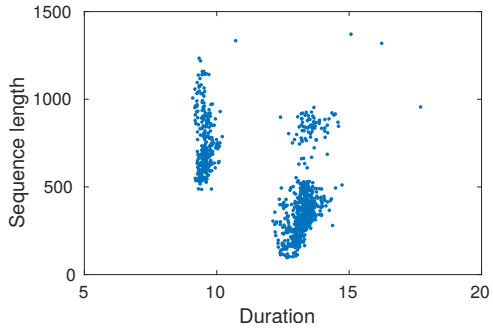
Appendix B

Supplementary figures

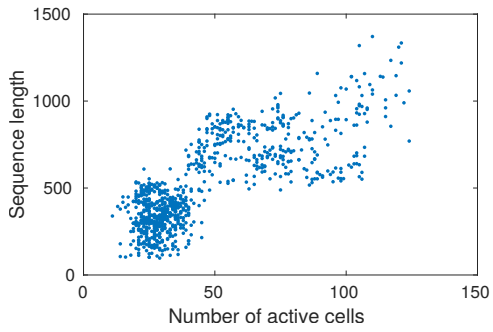


(a) Normal condition (b) Muscimol condition (c) Increase of occurrence

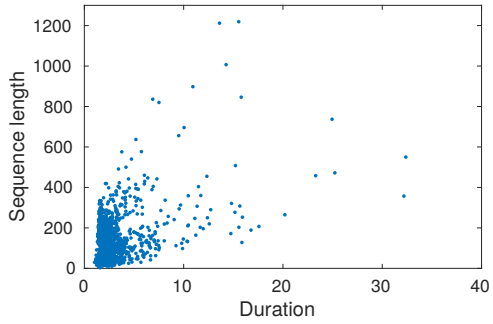
Figure B.1: SWR occurrence. Panel (a) shows that under normal conditions, SWR events occur almost exclusively at the reward areas, though some small number occur within the delay area. Panel (b) shows that injection of muscimol is accompanied by an increased number of SWRs occurring within the delay area. Panel (c) shows the ratio of the number of muscimol events to the number of normal-condition events on a per-recording basis.



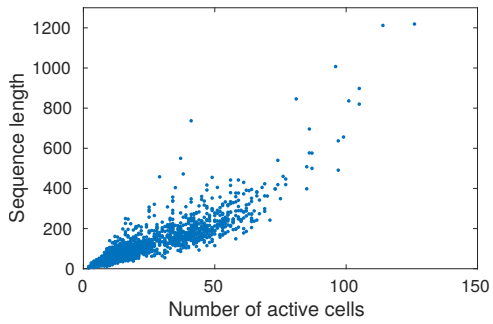
(a) Wheel



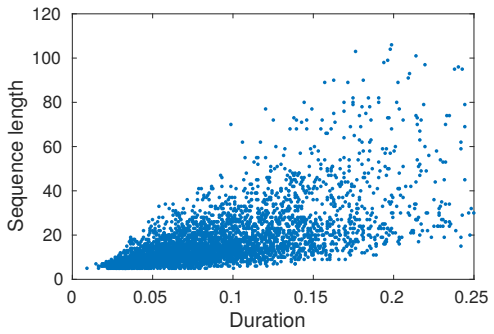
(b) Wheel



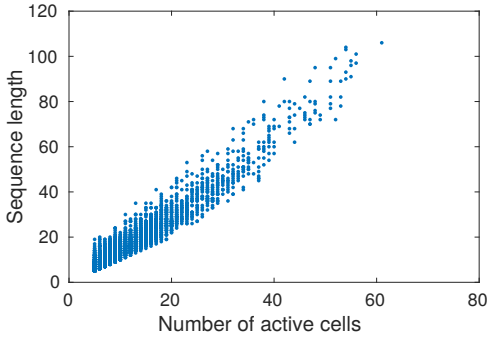
(c) Arm



(d) Arm

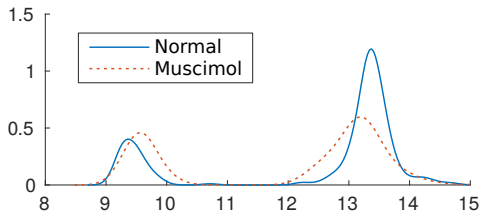


(e) SWR

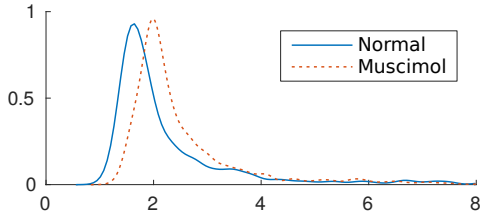


(f) SWR

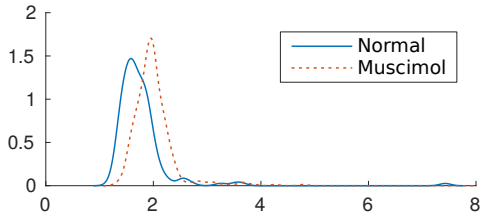
Figure B.2: Sequence length. The left column shows the relationship between the duration of an event and the corresponding sequence's length. The right column shows the relationship between the total number of neurons that are active within an event and the corresponding sequence's length. Each plot in the right column shows that a linear dependence exists between the number of active cells in an event and the total length of the sequence.



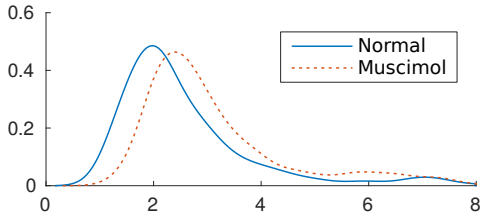
(a) **Wheel-run.** Median: 13.30 / 12.94; KS: 0.0046



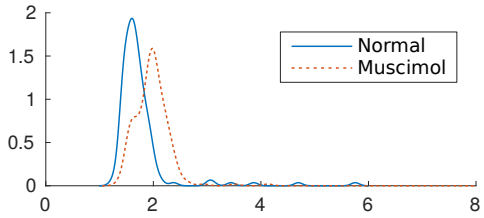
(b) **Arm-run.** Median: 1.84 / 2.16; KS: 1.



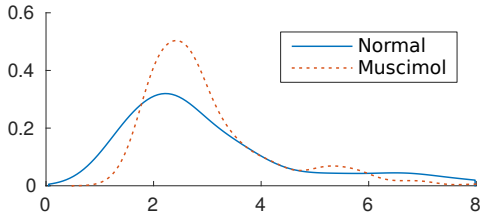
(c) **Left / outbound.** Median: 1.66 / 1.96; KS: 0.



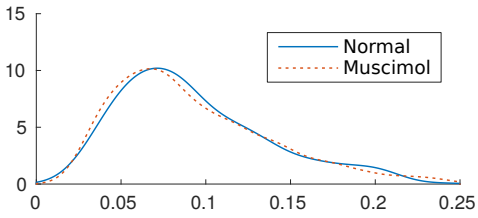
(d) **Left / inbound.** Median: 2.25 / 2.74; KS: 1.



(e) **Right / outbound.** Median: 1.64 / 1.97; KS: 8.



(f) **Right / inbound.** Median: 2.65 / 2.70; KS: 0.



(g) **Refined SWR.** Median: 0.084 / 0.082; KS: 2.99

Figure B.3: Event duration. For each event type, the distribution of event durations is shown under normal (solid line) and muscimol (dashed line) conditions. The median of each distribution is shown as normal / median. In each case, the two-sample Kolmogorov-Smirnov test (KS) tells us the p-value with with we reject the null hypothesis that the two distributions are the same. For each behavioral event type (all but SWR), the injection of muscimol tends to create slightly longer events. In contrast, muscimol tends to create slightly shorter SWR events.

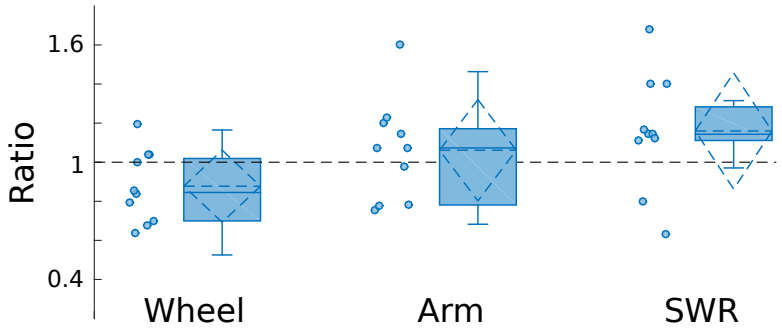


Figure B.4: Number of active neurons. The vertical axis shows the ratio of the median number of active neurons in a muscimol event to the median number of active neurons in an event under normal conditions. After injection of muscimol, the median number of active cells in wheel-run sequences tends to decrease slightly, whereas the median number of The median number of active neurons changes after. No change is observed in the number of active neurons in arm-run sequences.

Bibliography

[1] J. G. White et al. "The Structure of the Nervous System of the Nematode Caenorhabditis elegans". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 314.1165 (Nov. 1986), pp. 1–340. DOI: 10.1098/rstb.1986.0056. URL: http://dx.doi.org/10.1098/rstb.1986.0056.

[2] Frederico A.C. Azevedo et al. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain". In: *J. Comp. Neurol.* 513.5 (Apr. 2009), pp. 532–541. DOI: 10.1002/cne.21974. URL: http://dx.doi.org/10.1002/cne.21974.

[3] W. B. Scoville and B. Milner. "LOSS OF RECENT MEMORY AFTER BILATERAL HIPPOCAMPAL LESIONS". In: *Journal of Neurology, Neurosurgery & Psychiatry* 20.1 (Feb. 1957), pp. 11–21. DOI: 10.1136/jnnp.20.1.11. URL: http://dx.doi.org/10.1136/jnnp.20.1.11.

[4] J. O'Keefe and J. Dostrovsky. "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat". In: *Brain Research* 34.1 (Nov. 1971), pp. 171–175. DOI: 10.1016/0006-8993(71)90358-1. URL: http://dx.doi.org/10.1016/0006-8993(71)90358-1.

[5] W. E. Skaggs et al. "Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences." eng. In: *Hippocampus* 6.2

(1996), pp. 149–172. ISSN: 1050-9631. DOI: 10.1002/(SICI)1098-1063(1996)6:2<149::AID-HIPO6>3.0.CO;2-K. URL: http://www.ncbi.nlm.nih.gov/pubmed/8797016.

[6] George Dragoi and György Buzsáki. "Temporal Encoding of Place Sequences by Hippocampal Cell Assemblies". In: *Neuron* 50.1 (Apr. 2006), pp. 145–157. DOI: 10.1016/j.neuron.2006.02.023. URL: http://dx.doi.org/10.1016/j.neuron.2006.02.023.

[7] George Dragoi and Susumu Tonegawa. "Preplay of future place cell sequences by hippocampal cellular assemblies". In: *Nature* 469.7330 (Dec. 2010), pp. 397–401. DOI: 10.1038/nature09633. URL: http://dx.doi.org/10.1038/nature09633.

[8] Gabrielle Girardeau et al. "Selective suppression of hippocampal ripples impairs spatial memory". In: *Nature Neuroscience* 12.10 (Sept. 2009), pp. 1222–1223. DOI: 10.1038/nn.2384. URL: http://dx.doi.org/10.1038/nn.2384.

[9] Kamran Diba and György Buzsáki. "Forward and reverse hippocampal place-cell sequences during ripples". In: *Nature Neuroscience* 10.10 (Sept. 2007), pp. 1241–1242. DOI: 10.1038/nn1961. URL: http://dx.doi.org/10.1038/nn1961.

[10] Yingxue Wang et al. "Theta sequences are essential for internally generated hippocampal firing fields". In: *Nature Neuroscience* 18.2 (Dec. 2014), pp. 282–288. DOI: 10.1038/nn.3904. URL: http://dx.doi.org/10.1038/nn.3904.

[11] Michaël Zugaro. *Freely Moving Animal (FMA) Toolbox*. 2010. URL: http://fmatoolbox.sourceforge.net/ (visited on 07/06/2015).

[12] Lav R. Varshney et al. "Structural Properties of the Caenorhabditis elegans Neuronal Network". In: *PLoS Computational Biology* 7.2 (Feb. 2011). Ed. by

Olaf Sporns, e1001066. DOI: 10.1371/journal.pcbi.1001066. URL: http://dx.doi.org/10.1371/journal.pcbi.1001066.

[13] Masami Tatsuno, ed. *Analysis and Modeling of Coordinated Multi-neuronal Activity*. Springer New York, 2015. DOI: 10.1007/978-1-4939-1969-7. URL: http://dx.doi.org/10.1007/978-1-4939-1969-7.